

В ПОМОЩЬ РАДИОЛЮБИТЕЛЮ

Патрик Гёлль

ПК и чип-карты

Принципы работы и способы использования чип-карт, сопутствующий инструментарий и пакеты программ

ETSF

DUNOD

ДМК

В помощь радиолюбителю

Патрик Гёлль

Инженер EFREI

ПК и ЧИП-КАРТЫ



Москва, 2003

УДК 621.396.6
ББК 32.842-5я92
Г31

Гёлль П.

Г31 ПК и чип-карты: Пер. с фр. – М.: ДМК Пресс, 2003. – 144 с.: ил. (В помощь радиолюбителю).

ISBN 5-94074-200-2

Вторая книга известного французского инженера Патрика Гелля, посвященная чип-картам, позволяет читателю более углубленно изучить принципы работы и возможности использования в радиолюбительской практике самых современных чип-карт – с микропроцессором и перезаписываемых, которые описаны в книге «Чип-карты. Устройство и применение в практических конструкциях» (изд-во «ДМК», 2000 год). С другой стороны, это вполне самодостаточное издание, в котором подробно рассмотрены чип-карты с микропроцессорами, а также необходимые для работы с ними инструментарий и пакеты программ. Представленные конструкции отличает простота, дешевизна и технологичность.

Для всех устройств приводятся чертежи печатных плат и подробно описываются принципы работы схем.

Книга будет полезна как инженерам, желающим ознакомиться с новыми радиоэлементами, так и радиолюбителям, пытающимся найти применение использованным чип-картам.

ISBN 2-10-003886-9 (фр.)
ISBN 5-94074-202-5 (рус.)

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Когда в 1992 году была выпущена в свет книга “Cartes à puce, Initiation et applications”[1], вряд ли кто-либо предполагал, что спустя три года невинные опыты в области чтения и записи пустых телефонных карт приведут к проникновению в секреты настоящих «электронных крепостей», каковыми могут считаться карты на микропроцессорах и новые телефонные карты с памятью, данные в которых можно обновлять.

Изготовители и дистрибьюторы самих карт, уверенные в эффективности механизмов защиты их приложений, оказали автору любезное содействие в написании данной книги, и теперь появилась возможность показать читателям, как можно преобразовать ПК в мощное орудие исследования и даже имитации самых разных чип-карт.

Несколько простейших электронных схем и специальное программное обеспечение, представленное на сервере www.dmkpress.ru издательства «ДМК Пресс», дадут вам материал для захватывающих открытий. Счастливого пути в увлекательный мир чип-карт!

---

[1] Русский перевод: «Чип-карты. Устройство и применение в практических конструкциях», М.: ДМК, 2000. Файлы с программным обеспечением размещены на сайте www.dmkpress.ru.

МИКРОПРОЦЕССОРЫ, МИКРОКАЛЬКУЛЯТОРЫ ИЛИ МИКРОКОНТРОЛЛЕРЫ?

В 1981 году, за два года до внедрения первых чип-карт, у компании BULL появились первые однокристалльные чип-карты на микрокалькуляторах.

По тем временам это было большое открытие, достигнутое благодаря сотрудничеству с компанией Motorola, поскольку с момента возникновения первых прототипов, выпущенных в 1974 году для компании Roland Moreno, имело место сочетание по крайней мере двух различных чипов (микропроцессора и памяти).

В 1976 г. компания CII Honeywell-Bull приобрела лицензию у фирмы Innovatron и приняла термин «карта на микрокалькуляторе» для обозначения продукта, который до сегодняшнего дня составляет семейство CP8. Хотя конкуренты BULL CP8 чаще употребляют сочетание «карта на микропроцессоре», в наибольшей степени соответствует истине название «карта на микроконтроллере».

На самом деле архитектура современных чипов для карт объединяет центральный процессор на 8 бит, ПЗУ, ОЗУ, ППЗУ и/или ЭСППЗУ, порты ввода-вывода точно так же, как и все микроконтроллеры. Просто в настоящее время существует тенденция добавлять все больше так называемых ресурсов для обеспечения безопасности, размещаемых на том же чипе. Таковы, вероятно, перспективы развития чип-карт.

МАТЕРИАЛЬНЫЕ РЕСУРСЫ

Если очевидным лидером на рынке инкартируемых микроконтроллеров является компания Motorola, то среди прочих производителей, по достоинству оценивших этот бизнес, наблюдается ожесточенная конкуренция. Карты начинают выпускаться в астрономических количествах!

Компания SGS-Thomson занимает особенно удачные позиции со своей продукцией, основанной на ядре, совместимом с процессорами Motorola, и очень интересными решениями в области защиты. Philips решительно выходит на сцену с архитектурой 8051, внося существенный вклад в использование возможностей криптографических вычислений. Компания Texas Instruments, прямой конкурент SGS-Thomson на рынке чипов для телефонных карт, предлагает, со своей стороны, ряд инкартируемых микропроцессоров, выросших из семейства TMS370. И конечно, данным вопросом очень пристально интересуются японские производители, в первую очередь концерн OKI с его продукцией OSCAR.

Интересно в общих чертах сравнить блок-схемы микроконтроллеров различных марок, часто встречающихся в чип-картах. На рис. 1.1 воспроизводедена общая структурная схема всех продуктов Motorola, которые имеют в своей основе классическое ядро 68HC05.

Версией SC24 (3K ПЗУ, 1K ЭСППЗУ, 128 байт ОЗУ) снабжены, в частности, французские банковские карты.

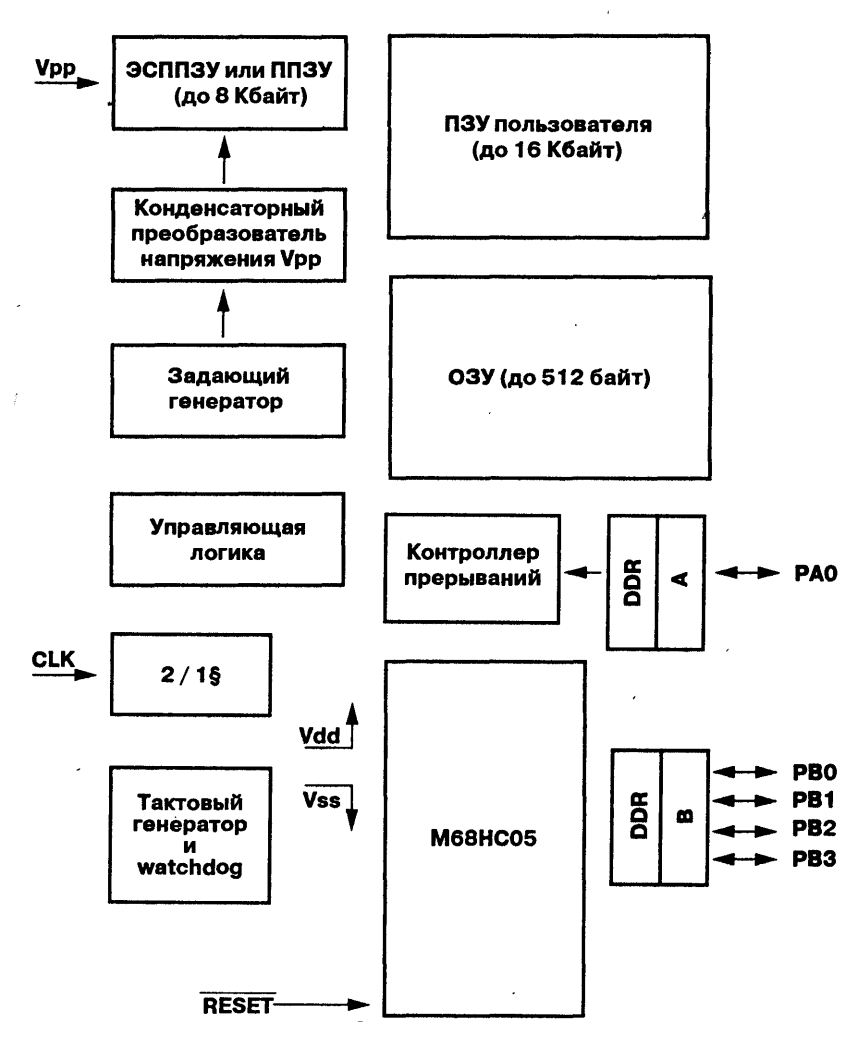


Рис. 1.1. Архитектура M68HC05SC Motorola

На рис. 1.2 показана архитектура кристаллов ST16XYZ, которая является «коньком» SGS-Thomson. Она объединяет совместимый с 6805 центральный процессор, наделенный рядом защитных функций, как аппаратных, так и программных: логическим блоком, который обеспечивает безопасность благодаря *датчикам*, определяющим попытки незаконного доступа к чипу; ПЗУ с *шифрованной шиной адреса*; системой маскировки колебаний тока питания при изменениях содержимого памяти; матрицей доступа, ограничивающей возможность несанкционированного доступа к памяти и т.д.

Отметим, что версией ST16301B на конкурентной основе снабжены французские банковские карты.

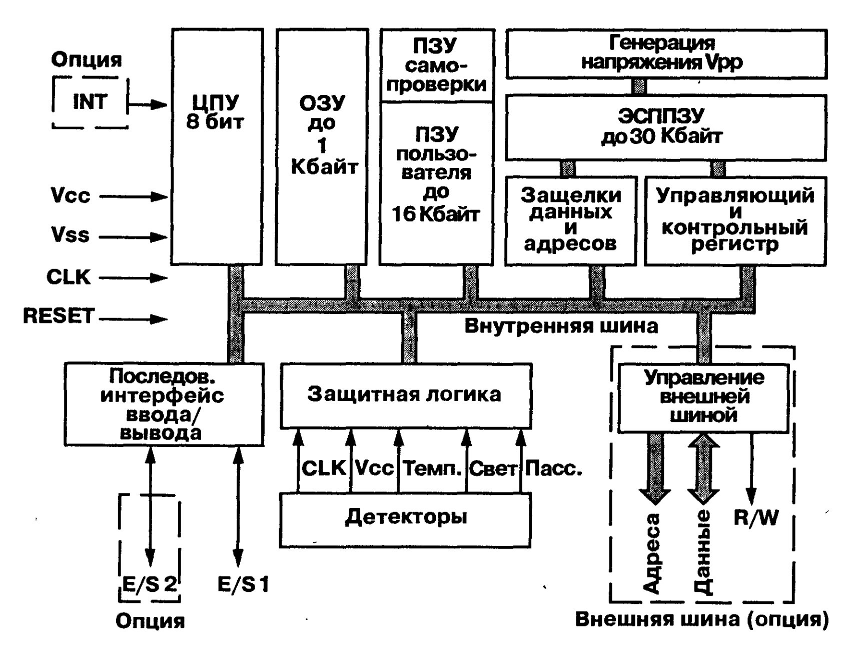


Рис. 1.2. Архитектура ST16XYZ SGS-Thomson

На рис. 1.3 продемонстрирован чрезвычайно прогрессивный процессор Philips, 83C852. Его ядру типа 8051 оказывает содействие специализированная *пересчетная система*, которая представляет собой настоящий быстродействующий сопроцессор, предназначенный для специальных операций, необходимых для кодирования данных. Все это содержится на одном чипе.

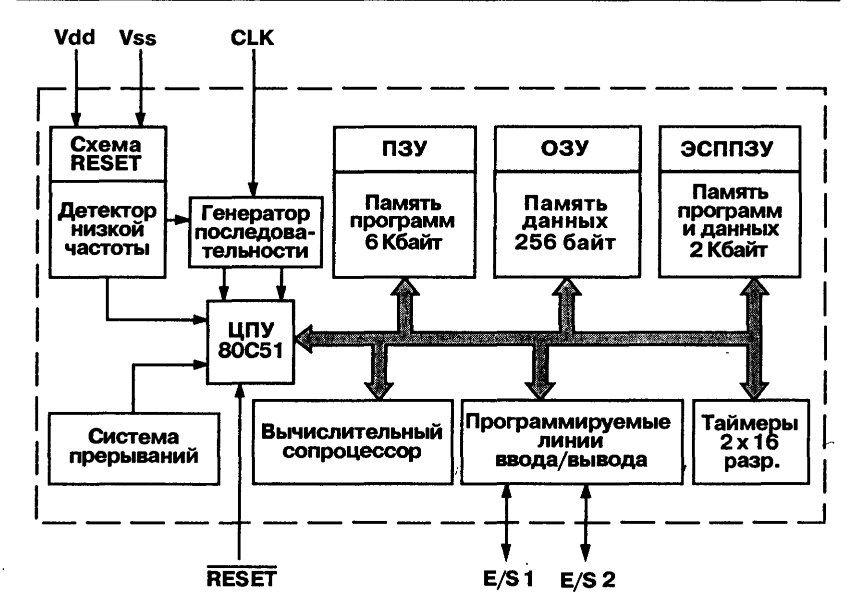


Рис. 1.3. Архитектура 83C852 Philips

Наконец, рис. 1.4 показывает внутреннюю организацию новейшего, но уже ставшего классическим чипа Texas Instruments – TMS373C007. Некоторые серии французских банковских карт снабжены чипом более поздней серии TMS373C012.

Можно отметить, что базовая структура карт разных марок почти не изменяется. Практически все разновидности (компоненты) располагают, например, двумя линиями последовательного ввода/вывода, хотя в настоящее время почти всегда используется только одна (*half-duplex*).

Ресурсы памяти, часто изменяемые в соответствии с требованиями клиента, практически всегда зависимы от одних и тех же технологических ограничений, накладываемых площадью кремния, которую можно удобно расположить на карте (приблизительно 4×6 мм). Правда, применение субмикронных технологий дает шанс улучшить ситуацию в относительно скором будущем.

К тому же многое можно расположить на 8 Кбайт ЭСППЗУ (перезаписываемого) или ППЗУ (применяемого без возможности повторного использования), а в 16 Кбайт ПЗУ удастся разместить код вполне достойной *операционной системы*.

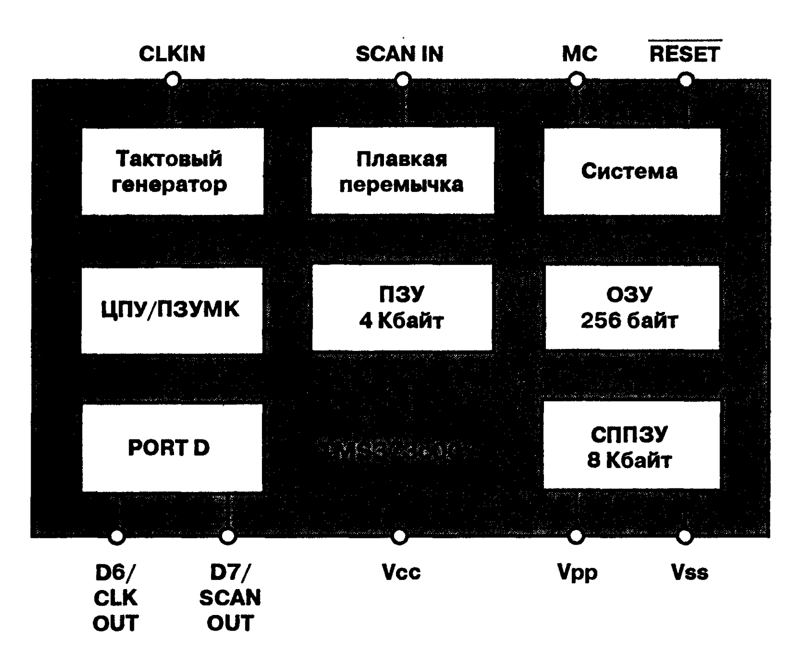


Рис. 1.4. Архитектура TMS373C007

ПРОГРАММНЫЕ РЕСУРСЫ

Все чипы карт на микроконтроллерах происходят напрямую от того или иного классического семейства микропроцессоров или микроконтроллеров. От последних они наследуют бóльшую часть системы команд, даже если принято удалять из них несколько функций, не представляющих особого интереса в области чип-карт; правда, вместо таких команд добавляются более специфические.

Компания Texas Instruments разработала команду NOVP (NOP с варьируемой длиной, очень полезную для выполнения операции выдержки времени), а также команду перемещения бит MOVB.

В табл. 1.1 воспроизводится полная система команд, наиболее широко распространенная в мире карт на микропроцессорах, так как она в значительной степени является общей для чипов SGS-Thomson и Motorola.

В принципе основные манипуляции с картами на микропроцессорах не нуждаются в прямом программировании центрального процессора (и даже не позволяют этого). С точки зрения удобства использования,

Таблица 1.1. Система команд для чип-карт ST16XYZ

Register/Memory and Absolute Jump Group –
группа Регистр/память и Безусловные переходы

Function	Mnemonic	Addressing Mode					
		Immediate	Direct	Extended	Index no offset	Index 8 bit offset	Index 16 bit offset
Load A with memory	LDA	$_2$A6^2	$_2$B6^3	$_3$C6^4	$_1$F6^3	$_2$E6^4	$_3$D6^5
Load X with memory	LDX	$_2$AE^2	$_2$BE^3	$_3$CE^4	$_1$FE^3	$_2$EE^4	$_3$DE^5
Load memory with A	STA		$_2$B7^3	$_3$C7^4	$_1$F7^3	$_2$E7^4	$_3$D7^5
Load memory with X	STX		$_2$BF^3	$_3$CF^4	$_1$FF^3	$_2$EF^4	$_3$DF^5
Add memory to A	ADD	$_2$AB^2	$_2$BB^3	$_3$CB^4	$_1$FB^3	$_2$EB^4	$_3$DB^5
Add memory and carry to A	ADC	$_2$A9^2	$_2$B9^3	$_3$C9^4	$_1$F9^3	$_2$E9^4	$_3$D9^5
Subtract memory to A	SUB	$_2$A0^2	$_2$B0^3	$_3$CO^4	$_1$F0^3	$_2$E0^4	$_3$D0^5
Subtract memory with carry	see	$_2$A2^2	$_2$B2^3	$_3$C2^4	$_1$F2^3	$_2$E2^4	$_3$D2^5
And memory to A	AND	$_2$A4^2	$_2$84^3	$_3$C4^4	$_1$F4^3	$_2$E4^4	$_3$D4^5
Or memory with A	ORA	$_2$AA^2	$_2$BA^3	$_3$CA^4	$_1$FA^3	$_2$EA^4	$_3$DA^5
Exclusive OR	EOR	$_2$A8^2	$_2$B8^3	$_3$C8^4	$_1$F8^3	$_2$E8^4	$_3$D8^5
Arithmetic Compare A	CMP	$_2$A1^2	$_2$B1^3	$_3$C1^4	$_1$F1^3	$_2$E1^4	$_3$DI^5
Arithmetic Compare X	CPX	$_2$A3^2	$_2$B3^3	$_3$C3^4	$_1$F3^3	$_2$E3^4	$_3$D3^5
Bit compare A and memory	BIT	$_2$A5^2	$_2$B5^3	$_3$C5^4	$_1$F5^3	$_2$E5^4	$_3$D5^5
Absolute Jump	JMP		$_2$BC^2	$_3$CC^3	$_1$FC^2	$_2$EC^3	$_3$DC^4
Jump to subroutine	JSR		$_2$BD^5	$_3$CD^6	$_1$FD^5	$_2$ED^6	$_3$DD^7

Bit manipulation and test Group – группа обработки и тестирования битов

Function	Mnemonic	Addressing Mode
Bit Set	BSET b (b=0..7)	$_2$(10+2*b)^5
Bit clear	BCLR b (b=0..7)	$_2$(11+2*b)^5
Test bit b and branch if true	BRSET b (b=0..7)	$_3$(00+2*b)^5
Test bit b and branch if false	BRCLR b (b=0..7)	$_3$(01+2*b)^5

Таблица 1.1. Система команд для чип-карт ST16XYZ (продолжение)

Read/Modify/Write Group – группа Чтение/Изменение/Запись

Function	Mnemonic	Addressing Mode				
		Inherent A	Inherent X	Direct	Index no offset	Index 8 bit offset
Increment	INC	${}_{1}4C^{2}$	${}_{1}5C^{2}$	${}_{2}3C^{5}$	${}_{1}7C^{5}$	${}_{2}6C^{6}$
Decrement	DEC	${}_{1}4A^{2}$	${}_{1}5A^{2}$	${}_{2}3A^{5}$	${}_{1}7A^{5}$	${}_{2}6A^{6}$
Clear	CLR	${}_{1}4F^{2}$	${}_{1}5F^{2}$	${}_{2}3F^{5}$	${}_{1}7F^{5}$	${}_{2}6F^{6}$
One's Complement	COM	${}_{1}43^{2}$	${}_{1}53^{2}$	${}_{2}33^{5}$	${}_{1}73^{5}$	${}_{2}63^{6}$
Negate (2's Complement)	NEG	${}_{1}40^{2}$	${}_{1}50^{2}$	${}_{2}30^{5}$	${}_{1}70^{5}$	${}_{2}60^{6}$
Rotate Left thru Carry	ROL	${}_{1}49^{2}$	${}_{1}59^{2}$	${}_{2}39^{5}$	${}_{1}79^{5}$	${}_{2}69^{6}$
Rotate Right thru Carry	ROR	${}_{1}46^{2}$	${}_{1}56^{2}$	${}_{2}36^{5}$	${}_{1}76^{5}$	${}_{2}66^{6}$
Logical Shift Left into Carry	LSL	${}_{1}48^{2}$	${}_{1}58^{2}$	${}_{2}38^{5}$	${}_{1}78^{5}$	${}_{2}68^{6}$
Logical Shift Right into Carry	LSR	${}_{1}44^{2}$	${}_{1}54^{2}$	${}_{2}34^{5}$	${}_{1}74^{5}$	${}_{2}64^{6}$
Arithmetic Shift Right into Carry	ASR	${}_{1}47^{2}$	${}_{1}57^{2}$	${}_{2}37^{5}$	${}_{1}77^{5}$	${}_{2}67^{6}$
Test for Negative or Zero	TST	${}_{1}4D^{2}$	${}_{1}5D^{2}$	${}_{2}3D^{3}$	${}_{1}7D^{3}$	${}_{2}6D^{4}$

Branch Group – группа ветвления (условных переходов)

Function	Mnemonic	Addressing Mode RELATIVE
Branch Always	BRA	${}_{2}20^{3}$
Branch Never	BRN	${}_{2}21^{3}$
Branch if Higher	BHI	${}_{2}22^{3}$
Branch if Unsigned Lower or Same	BLS	${}_{2}23^{3}$
Branch if Carry Clear	BCC	${}_{2}24^{3}$
Branch if Unsigned Higher or Same	BHS	${}_{2}24^{3}$
Branch if Carry Set	BCS	${}_{2}25^{3}$
Branch if Unsigned Lower than	BLO	${}_{2}25^{3}$
Branch if Not Equal	BNE	${}_{2}26^{3}$
Branch if Equal	BEQ	${}_{2}27^{3}$
Branch if Half Carry Clear	BHCC	${}_{2}28^{3}$
Branch if Not Half Carry Set	BHCS	${}_{2}29^{3}$
Branch if Plus	BPL	${}_{2}2A^{3}$
Branch if Minus	BMI	${}_{2}2B^{3}$
Branch if Not Interrupt Mask	BMC	${}_{2}2C^{3}$
Branch ff Interrupt Mask	BMS	${}_{2}2D^{3}$
Branch if Interrupt Line Low	BIL	${}_{2}2E^{3}$
Branch if Interrupt Line High	BIH	${}_{2}2F^{3}$
Branch to Subroutine	BSR	${}_{2}AD^{5}$

Таблица 1.1. Система команд для чип-карт ST16XYZ (окончание)

Miscellaneous Group – дополнительная группа

Function	Mnemonic	Addressing Mode
		INHERENT
Multiply (X : A=X* A)	MUL	${}_{1}42^{10}$
Transfer A to X	TAX	${}_{1}97^{2}$
Transfer X to A	TXA	${}_{1}9F^{2}$
Transfer SP to A	TSA	${}_{1}9E^{2}$
Clear Carry Flag	CLC	${}_{1}98^{1}$
Set Carry Flag	SEC	${}_{1}99^{1}$
Clear Interrupt Mask bit	CLI	${}_{1}9A^{2}$
Set Interrupt Mask bit	SEI	${}_{1}9B^{2}$
Reset Stack Pointer	RSP	${}_{1}9C^{2}$
No Operation	NOP	${}_{1}90^{2}$
Return from Interrupt	RTI	${}_{1}80^{2}$
Return from Subroutine	RTS	${}_{1}81^{s}$
Software Interrupt	SWI	${}_{1}83^{9}$
Halt CPU/Enable INT	WAIT	${}_{1}8F^{2}$
Halt CPU/STOP Clocks/Enable INT	STOP	${}_{1}8E^{2}$

а также и безопасности, процессор полностью находится под контролем операционной системы, расположенной в ПЗУ.

Есть, впрочем, исключение, лишь подтверждающее правило: некоторые карты, так называемые COS, обладают своеобразным *портом ввода*, позволяющим «имплантировать» некоторый объем собственного кода в ППЗУ или в ЭСППЗУ. В этом случае необходимо хорошее знание системы команд процессора. Именно эти карты часто снабжены чипом ST16CXYZ.

РЕСУРСЫ, ОБЕСПЕЧИВАЮЩИЕ БЕЗОПАСНОСТЬ

Каким бы ни было разрабатываемое приложение, назначение чип-карты можно всегда свести к сохранению данных в более или менее защищенной форме – ведь речь идет о картах с памятью.

Превосходство карты на микропроцессоре по отношению к обычной карте с памятью, даже защищенной, обусловлено, с одной стороны, более простым и упорядоченным протоколом связи, а с другой – способностью проводить сложную обработку данных, которые карта принимает или передает.

Благодаря *криптографическим ресурсам* карта на микропроцессоре может не дать двух идентичных ответов на один и тот же запрос,

по крайней мере, если правильно использовать возможности, предлагаемые ее операционной системой. Очевидно, что подобная организация в высшей степени усложняет всякую попытку *повторного воздействия*, то есть перехвата и последующего воспроизведения ответа карты.

Однако, даже если принять во внимание вышесказанное, некоторые секретные сведения, содержащиеся в памяти, ни в коем случае не должны покидать карту: в идеале только операционная система имеет возможность доступа к ним, используя их в качестве элементов расчетов. Притом в открытом виде может появиться только результат. Подобным образом действуют *конфиденциальные коды*, которые присваиваются владельцу карты (*транспортный код, ключ владельца*) или ее дистрибьютору: нет никакой возможности прочитать их, как с простой магнитной полосы, чтобы сравнить вне карты с кодом, набранным на клавиатуре!

Процедура обеспечения безопасности состоит в том, чтобы предъявить код карте, которая будет проверять его с помощью своей операционной системы, внутри. После проведения такой проверки карта может информировать (открыто или закодированным способом) о том, что код правилен или ложен, или довольствоваться выдачей разрешения на доступ в зону памяти, которая до этого была закрыта. Столь высокий уровень безопасности позволяет использовать карты или ключи на микропроцессоре в финансовой сфере (банковские карты, электронные кошельки и т.д.), области сотовой связи (телефоны GSM) или платного телевидения.

На рис. 1.5 показывается (согласно идее SGS-Thomson), как построены телевизионные декодеры, совместимые с наиболее современными кодирующими системами.

Снабженная памятью ЭСППЗУ с содержимым, подлежащим многократным изменениям, карта содержит все секретные данные, которые, наряду с получаемыми во время передачи, необходимы микропроцессору декодера для управления своим цифровым декодирующим устройством. Информация, содержащаяся в карте, может изменяться дистанционно с помощью обычного телевизионного передатчика посредством команд, включаемых в видеосигнал. В случае глубоких изменений в системе намного практичнее рассылать по почте новые карты, чем заниматься обменом декодеров.

Интересно отметить, что чипы, используемые в этой системе, являются гораздо более мощными, чем применяемые в банковских картах!

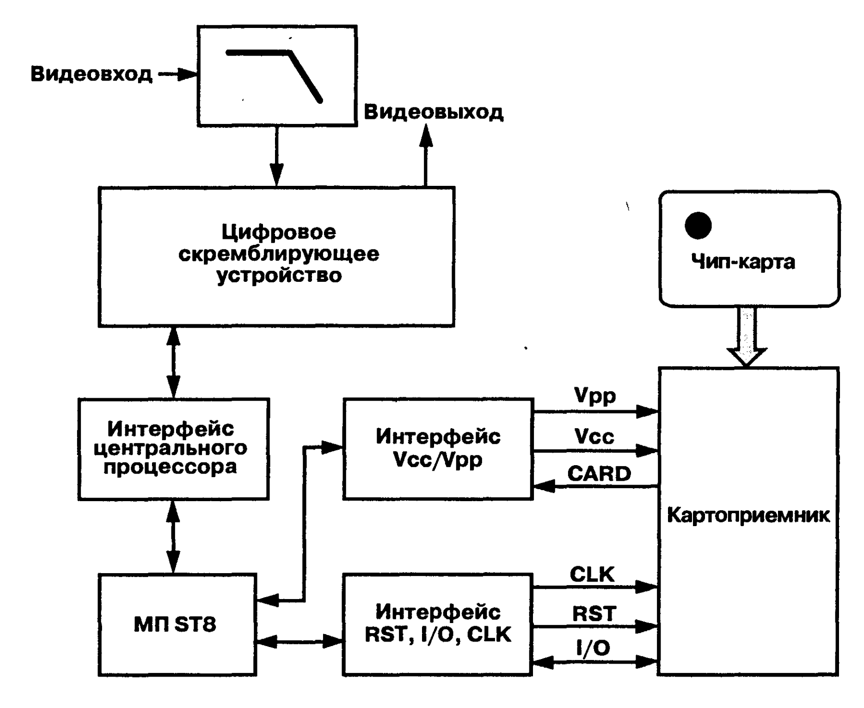


Рис. 1.5. Блок-схема телевизионного декодера с картой

НЕСКОЛЬКО ПОНЯТИЙ ИЗ КРИПТОГРАФИИ

Цель кодирования (науки составления секретных сообщений) заключается в возможности передачи конфиденциальных сведений в форме сообщения, лишенного какого-либо видимого смысла, на случай, если кто-нибудь перехватит послание. Для этого приступают к шифрованию данных с помощью ключа, затем к их дешифрованию посредством того же самого ключа (так называемый симметричный алгоритм) или другого (асимметричный алгоритм).

Очевидно, что если ключ раскодирования идентичен ключу кодирования, его защита предполагает те же проблемы, что и защита данных. На самом деле можно, конечно, расшифровать сообщение, если знать ключ и алгоритм кодирования. Но это вероятно и при полном незнании алгоритма, если имеются ключ и образцы некодированных сообщений в сопровождении их кодированной версии.

Одна из наиболее простых процедур шифрования в информатике – проведение операции Исключающее ИЛИ между каждым байтом

кодируемых данных и каждым байтом ключа. Данный метод интересен постольку, поскольку один и тот же алгоритм, применяемый с одинаковым ключом, позволяет проведение операции декодирования.

При условии безупречной защиты ключа и его однократного использования даже такая простейшая процедура обеспечивает абсолютную защиту в случае, если длина ключа (число байтов) будет, по крайней мере, равна длине кодируемого сообщения. Если напротив, один и тот же ключ небольшой длины неоднократно используется для кодирования различных частей гораздо более длинного сообщения, это ставит под сомнение всю надежность шифровки.

Небольшая программа XOR.BAS позволяет познакомиться с этим элементарным методом кодирования, правда, в упрощенном виде: ключ состоит из единственного байта, то есть одна буква всегда выражается одним и тем же числом (это соответствует шифрованию замещением, аналогичному той системе тайнописи, которую используют скауты). Нельзя не согласиться, что если длина сообщения ограничивается одним байтом, то совершенно невозможно обойтись без ключа для его раскодирования.

```
10 REM -- XOR.BAS --
20 A$="(c)1995 Патрик Гелль"
30 FOR G=1 TO LEN(A$)
40 D$=MID$(A$,G,1):D=ASC(D$)
50 R=D XOR 47
55 REM общий ключ = 47
70 PRINT D$,R,
80 K=R XOR 47
100 PRINT CHR$(K)
110 NEXT G
120 END
```

Проблема ключей была очень искусно решена по так называемому принципу алгоритмов на открытых ключах. Подобная криптографическая система использует два различных ключа: один для кодирования и другой для декодирования. В общем случае ключ кодирования является *открытым*, а декодирования – *секретным*: для того, чтобы отправить кому-либо сообщение, у адресата узнают открытый код (в крайнем случае его разыскивают в специальном справочнике).

Даже если алгоритмы кодирования и декодирования являются открытыми, расшифровать закодированные сообщения может только лицо, обладающее секретным кодом, с помощью своего открытого ключа. Но система работает и по обратному принципу: сообщение, зашифрованное с помощью секретного ключа, может быть расшифровано любым владельцем соответствующего открытого ключа, который

таким образом будет иметь формальное доказательство происхождения сообщения (речь идет об электронной подписи).

Наиболее известным алгоритмом на открытом ключе является алгоритм RSA (Rivest Shamir Adleman), который использует математические свойства степеней по модулю N: выбираются два простых числа (p и q), произведение которых (pq) послужит модулем для вычисления последующих степеней. Проще говоря, A в степени B по модулю N представляет собой A, умноженное B раз на A, минус столько раз N, чтобы получить положительный результат, меньший чем модуль N.

Алгоритм RSA основывается на том факте, что

$(p-1)(q-1)/x = 1$ по модулю pq

при условии, что x не делится ни на p, ни на q.

На практике это условие выполняется автоматически, если каждое из чисел p и q больше x, что, в общем, характерно при кодировании (чаще всего работают на числах разрядностью 512 бит).

```
10 REM -- MODULO.BAS --
20 KEY OFF:CLS
30 INPUT"Данные? ",D
40 INPUT"Показатель степени? ",E
50 INPUT"Коэффициент? ",M
60 R=D
70 FOR F=1 TO E-1
80 R=R*D
90 IF R<M THEN 110
100 R=R-M:GOTO 90
110 NEXT F
120 PRINT"результат: ";R
130 PRINT:GOTO 30
140 REM (c)1995 Patrick GUEULLE
```

Небольшая программа MODULO.BAS позволит провести несколько экспериментов со свободно выбранными операндами, а кроме этого проиллюстрирует принцип, используемый для возведения в степень по модулю с помощью программы на языке BASIC, без особого риска возникновения ошибок переполнения (overflow).

Оба ключа RSA (открытый e и секретный d или наоборот) выбираются так, чтобы они отвечали условию:

$ed = 1$ модуль$(p-1)(q-1)$.

В таком случае мы имеем:

$(x^d)^e = x$(модуль pq)

При этом ничто не позволяет вывести d из e или наоборот.

```
10 REM -- RSA.BAS --
20 A$="(c)1995 Патрик Гелль"
30 FOR G=1 TO LEN(A$)
40 D$=MID$(A$,G,1):D=ASC(D$)
50 E=15:M=391
55 REM Открытый ключ = 15, коэффициент = 391
60 GOSUB 130
70 PRINT D$,R,
80 E=47:M=391
85 REM Секретный ключ = 47, коэффициент = 391
90 D=R:GOSUB 130
100 PRINT CHR$(R)
110 NEXT G
120 END
130 R=D
140 FOR F=1 TO E-1
150 R=R*D
160 IF R<M THEN 180
170 R=R-M:GOTO 160
180 NEXT F
190 RETURN
```

Программа RSA.BAS применяет данный алгоритм в условиях, полностью сравнимых с вышеназванными, при использовании следующих ключей:

- открытый ключ: 15, модуль 391;
- секретный ключ: 47, модуль 391.

Значения получены из простых чисел p = 17 и q = 23.

Конечно, надежность операции не выше, чем в предыдущем случае, так как обработка идет байт за байтом, в то время как лучше было бы обрабатывать блоки по 512 бит. Однако преимущество этого способа в том, что он показывает, насколько замедлены вычисления даже при работе с блоками по 8 бит: именно благодаря этому ныне возможно снабжать чип-карты алгоритмом RSA только при наличии на карте или на чипе специального сопроцессора, пока не появились «крутые» чип-карты с 32-битным ядром. Например, с микроконтроллером 83C852 удается провести вычисления за полторы секунды, в то время как для проведения аналогичной операции с помощью базовой системы команд микропроцессора на 8 бит понадобилось бы почти три минуты. По этой причине большинство карт на микропроцессорах довольствуется симметричным алгоритмом, ультрасекретный ключ которого физически присутствует на карте и в связи с этим должен очень тщательно защищаться.

Наиболее широко используется американский алгоритм DES (Data Encryption Standard), который оперирует блоками по 64 бита. Также можно упомянуть TELEPASS, алгоритм, разработанный для чип-карт BULL CP8.

ПОНЯТИЕ МАСКИ

В микроэлектронике термином «маска» обозначаются фотошаблоны, служащие для проведения фотолитографических операций, необходимых для создания интегральных схем. В области чип-карт маской часто называют программное обеспечение операционной системы, встроенной в ПЗУ, которая определяет поведение карт на микропроцессорах. Это вполне логично, поскольку такое ПЗУ представляет собой *масочный тип*, то есть запрограммированный во время самого процесса создания чипов с помощью одной или нескольких специальных масок фотолитографии.

В подавляющем большинстве случаев фирма-изготовитель неизменно снабжает кристаллы для карт на микропроцессорах (и особенно сами карты) операционной системой. Поэтому владелец карты не в состоянии записать в нее свой собственный код на ассемблере, хотя для этого может быть использован порт ввода, который, впрочем, в любом случае находится под контролем операционной системы (точки входа, прерывания и т.д.).

В следующей главе будет объяснено, как создать свою карту на микроконтроллере, а также персональную мини-операционную систему.

МАСКИ BULL CP8

Наиболее распространенная карта BULL CP8, без сомнения, – M4, версия которой B0 представляет собой французскую банковскую чип-карту. Однако новое поколение уже начинает «принимать эстафету» в виде новой маски B0', являющейся собственностью банковского сообщества. Подробнее об этом будет рассказано в следующей главе.

Основываясь на данном базовом продукте, BULL CP8 последовательно разработала целое семейство карт на микропроцессорах. Их эволюция наглядно показана на рис. 1.6.

Отныне большинство небанковских приложений должно разрабатываться в рамках семейства SCOT. На рис. 1.7 показан его состав: от 8 до 64 Кбит (то есть от 1 до 8 Кбайт) памяти ППЗУ или ЭСППЗУ со встроенным алгоритмом кодирования TELEPASS или DES.

В целом карта CP8 представляет собой энергонезависимую память, объединенную на одном кристалле с микропроцессором, который полностью контролирует доступ к ней.

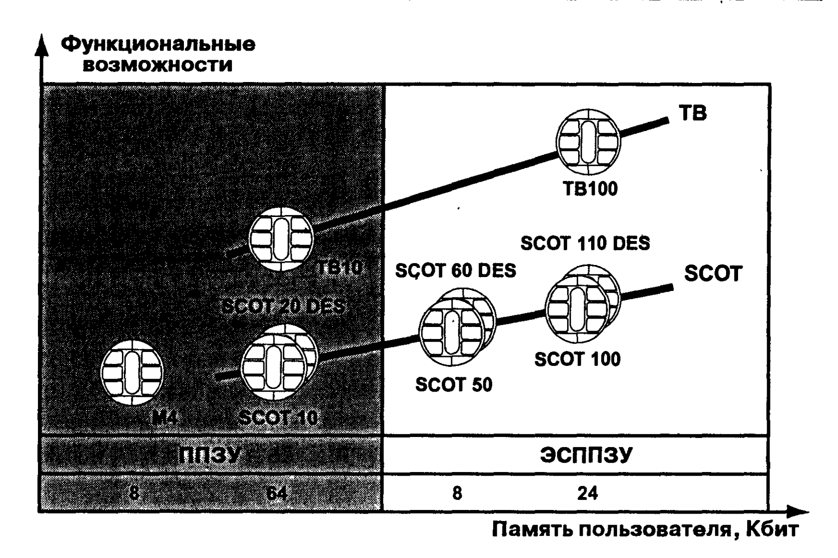


Рис. 1.6. Семейство карт CP8

Рис. 1.7. Карты BULL CP8 (с 6 и 8 контактами)

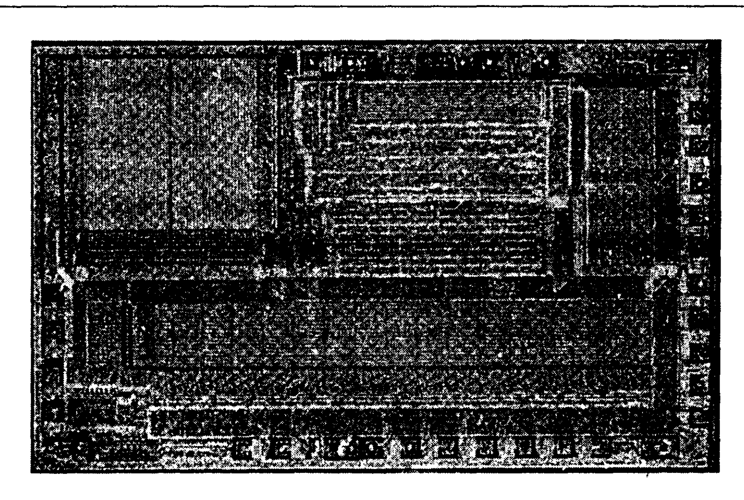


Рис. 1.8. Чип Motorola карты BULL CP8 (приблизительно 4x5 мм)

Наиболее секретные зоны могут быть прочитаны, записаны, или, при необходимости, стерты только с помощью микропроцессора; это совершенно невозможно сделать извне. Их содержание можно использовать только косвенным образом, например интерпретируя результат криптографического вычисления, в котором он представляет собой лишь один из параметров.

Память карт SCOT или M4 организована в виде небольшого числа зон с четко определенными прерогативами. На рис. 1.10 показана ее структура; видно, что степень защиты данных снижается по мере приближения к вершине памяти. Права доступа к различным зонам зависят от внутренней операционной системы карты, характера приложения, а также от стадии жизненного цикла карты (на заводе, у дистрибьютора, у владельца).

Табл. 1.4 определяет условия доступа к каждой из этих зон: от секретной, доступной исключительно микропроцессору, до зон чтения и изготовления, которые могут быть свободно считаны любым пользователем. Естественно, сам разработчик приложения решает, в какой зоне расположить ту или иную информацию, в зависимости от требуемого уровня безопасности.

Поскольку крайне важна возможность записывать информацию в наиболее секретных зонах во время изготовления карты и затем

у дистрибьютора в момент ее персонализации, четыре ключа, изображенные на рис. 1.9, могут быть необратимо введены в действие в конце каждой фазы жизненного цикла карты: первый ключ (LF) – до того как чип, индивидуально пронумерованный, покинул завод, а второй (LC) – когда дистрибьютор заканчивает свою работу по персонализации.

Два других ключа необязательно будут использоваться в течение нормального жизненного цикла карты: LU служит для нейтрализации изначального конфиденциального кода после его замены новым (в основном по инициативе владельца карты), в то время как IV позволяет полностью заблокировать карту, например в случае обнаружения попытки ее незаконного использования.

Таблица 1.2. Основные характеристики карт SCOT BULL CP8

SCOT 10	SCOT 20 DES	SCOT 50	SCOT 60 DES	SCOT 100	SCOT 110 DES
64 Кбит СППЗУ		8 Кбит ЭСППЗУ		24 Кбит ЭСППЗУ	
TELEPASS	DES	TELEPASS	DES	TELEPASS	DES
1	2	1	2	1	2

Таблица 1.3. Определение зон памяти карт BULL CP8

0200h	Секретная зона
ADM	Зона доступа
ADC	Конфиденциальная зона или рабочая зона № 2
ADT	Рабочая зона № 1
ADL	Зона чтения
ADMAX-8h	Зона изготовления

Таблица 1.4. Права доступа к зонам памяти карт BULL CP8

	Право доступа к различным зонам		
	Стирание	**Считывание**	**Запись**
Секретная зона	Запрещено	Запрещено	Запрещено
Зона доступа	Запрещено	Шифр	Запрещено
Конфиденциальная зона (опция «1 рабочая зона»)	Запрещено	Шифр	Запрещено
Рабочая зона 2 (опция «2 рабочих зоны»)	Приложение определяет права доступа к этой зоне		
Рабочая зона 1	Приложение определяет права доступа к этой зоне		
Зона считывания	Запрещено	Разрешено	Запрещено
Зона изготовления	Запрещено	Разрешено	Запрещено (кроме замков)

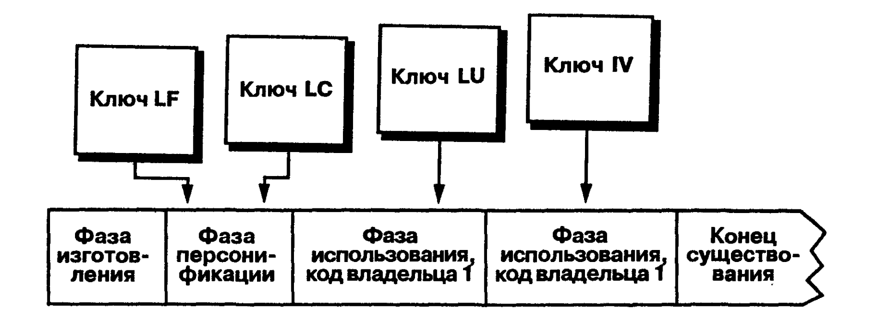


Рис. 1.9. Различные ключи карт BULL CP8

Каждая зона памяти организована словами по 32 бита (то есть 4 байта), адресуемых по уровню каждого ниббла (каждый адрес, таким образом, указывает на слово из 4 бит, а не на целый байт, поэтому для перехода к следующему байту необходимо двигаться через два адреса).

Важно отметить, что эти адреса необязательно соответствуют (практически никогда не соответствуют) тем, которые непосредственно обрабатывает микропроцессор. На самом деле адреса видны с точки зрения операционной системы, которая может их перемещать (или смешивать) так, как считает нужным.

Адрес начала каждой зоны содержится в специальном регистре-указателе, за исключением секретной зоны, которая начинается всегда с 0200. Таким образом, ADM отмечает начало зоны доступа, ADC – начало конфиденциальной зоны (или второй зоны работы), ADT – начало зоны работы, ADL – начало зоны чтения.

Зона изготовления расположена над последним словом памяти. Она заканчивается адресом ADMAX-8h, где ADMAX является последним адресом памяти, который может быть различным у разных карт.

Рис. 1.10 проясняет назначение этих зон для семейства SCOT. На рисунке показано, что регистры-указатели располагаются в зоне чтения – иными словами, полностью открыты, причем разработчик располагает широкими возможностями определения размера той или иной зоны.

Кроме этого, необходимо отметить, что ключи и конфиденциальные коды в обязательном порядке располагаются в секретной зоне, так же как и секретная система кодирования, используемая при

криптографических вычислениях. Следовательно, их невозможно будет прочитать даже на заводе.

Доступ к памяти производится под контролем микропроцессора при помощи системы команд, соответствующих стандарту 7816 (табл. 1.5). В соответствии с этим стандартом команда, отправляемая на карту устройством чтения-записи, представляется в виде блока из 5 байт, составленного следующим образом:

- так называемый байт класса равный BCh (для карт BULL CP8);
- байт, содержащий код операции системы команд;
- два байта, уточняющих адрес, к которому должна обращаться команда;
- байт, обозначающий длину блока данных, который должен быть отправлен на карту или получен от нее (00h, если нет обмена данными).

Таблица 1.5. Система команд для карт семейства SCOT

Команда	Порядок выполнения команд
B0h	Считывание
D0h	Запись
A0h	Поиск слова по аргументу
A0h	Поиск первого незаполненного слова
A8h	Поиск первого заполненного слова
C0h	Считывание результата
50h	Запись замков
0Eh	Стирание области
C4h	Запрос ошибочного числа
10h 30h 20h	Представление в открытом виде ключа или кода
20h 28h	Представление ключа разблокировки
18h 38h	Закодированное представление ключа
40h	Утверждение ключа или кода
70h	Утверждение к записи
80h	Вычисление сертификата
D2h	Изменение кода владельца

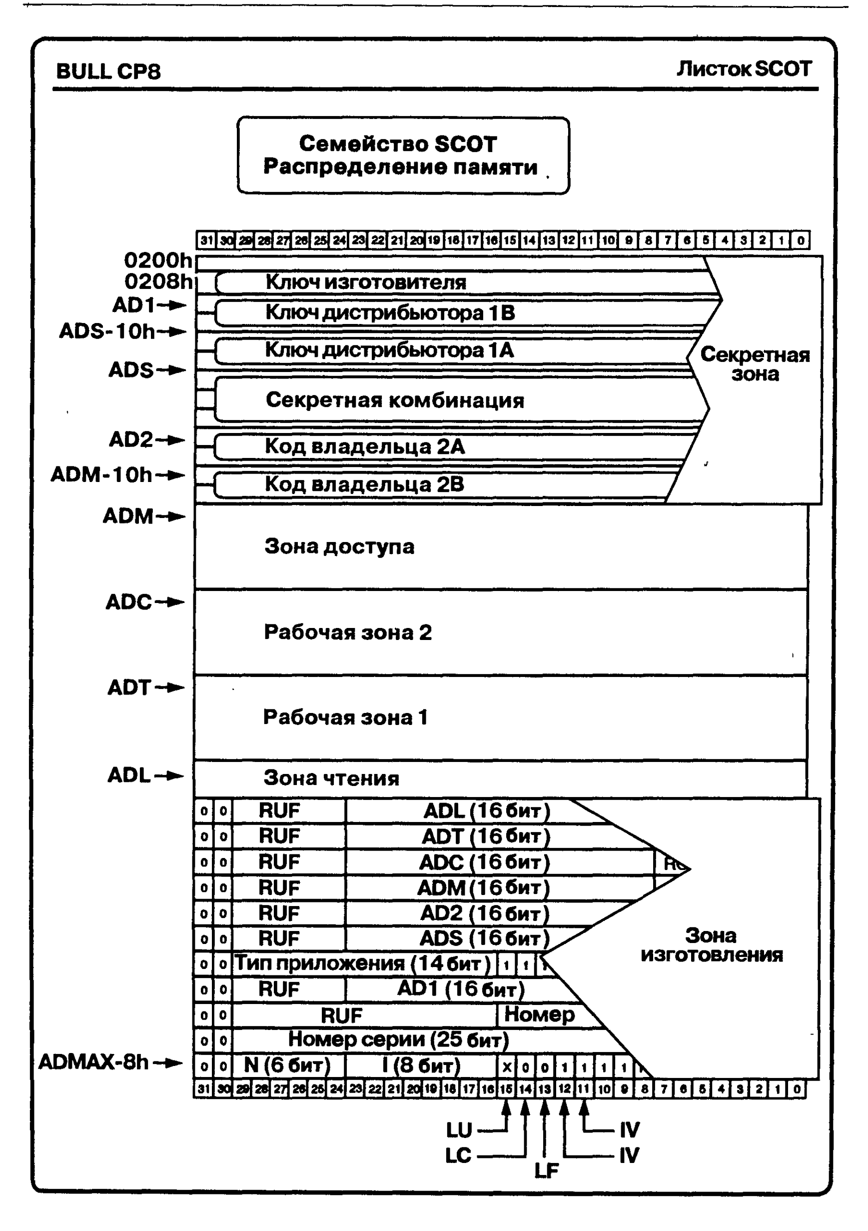


Рис. 1.10. Картография памяти карт SCOT

МАСКИ COS

Карты COS входят в разряд наиболее распространенных карт на микропроцессорах, что объясняется их исключительной гибкостью с точки зрения персонализации. COS расшифровывается как «операционная система чипа» (Chip Operating System) или «операционная система карты» (Card Operating System); имеется в виду операционная система, ответственная за управление всей памятью под защитой различных механизмов безопасности.

Рис. 1.11 представляет архитектуру системы, используемую в карте, которая существует в версии ППЗУ и ЭСППЗУ. Принцип ППЗУ, наиболее старый, не позволяет проводить операции стирания и повторного использования зон памяти: в этом случае рано или поздно может произойти насыщение или переполнение карты. Карты COS на ЭСППЗУ могут повторно использоваться почти до бесконечности.

Эта технология начинает применяться все шире, поскольку карту на микропроцессоре не нужно часто заменять.

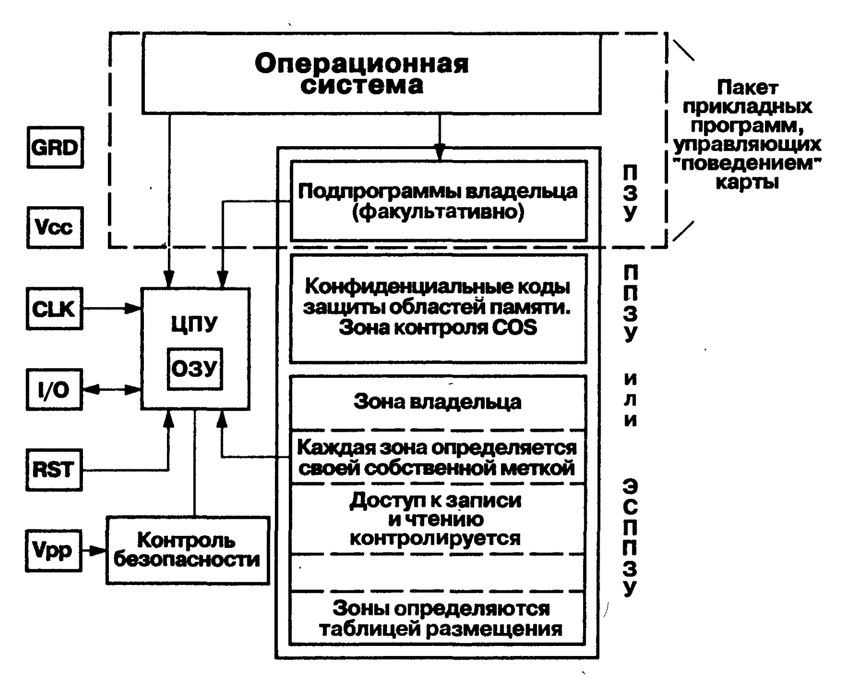


Рис. 1.11. Принцип действия маски COS

Вариант карты COS, так называемой MCOS, даже позволяет размещать на одной и той же карте несколько независимых приложений – в этом случае речь идет о многорежимной карте.

Рис. 1.12 представляет распределение памяти карт COS с 16 Кбайт ЭСППЗУ компании Gemplus Card International, в то время как рис. 1.13

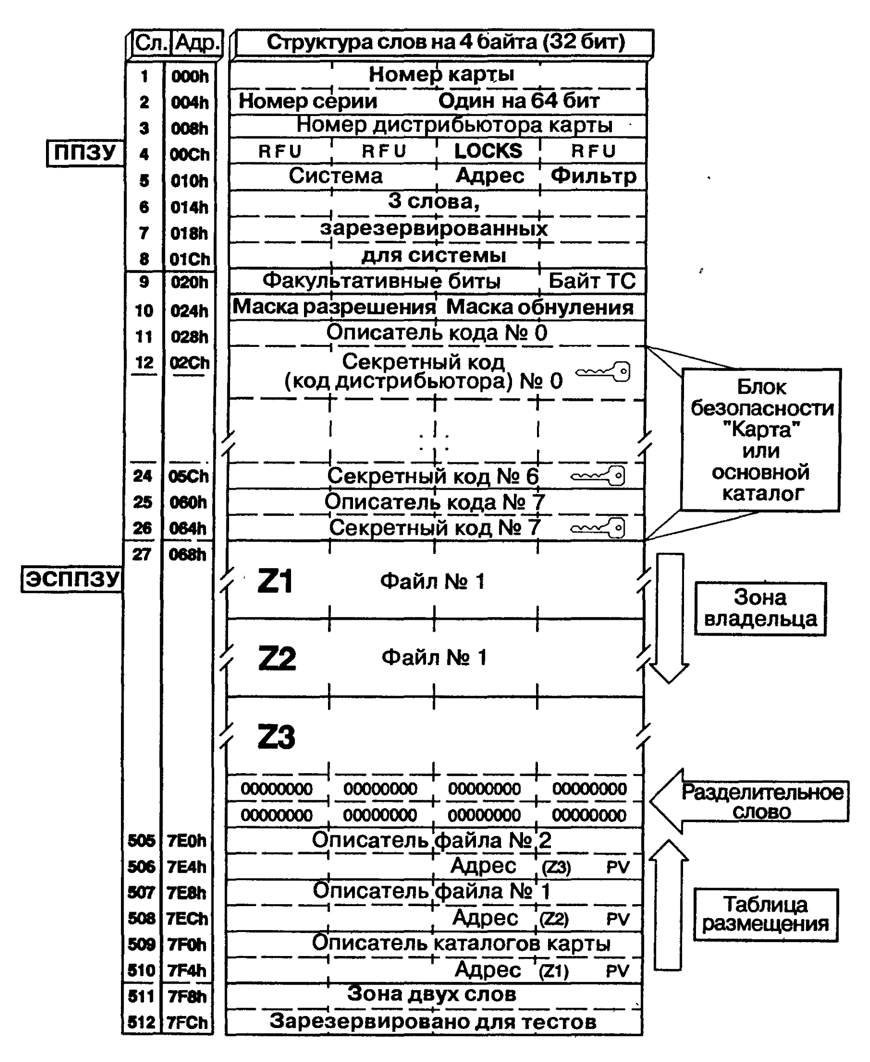


Рис. 1.12. Картография памяти карт COS на 16 Кбайт ЭСППЗУ

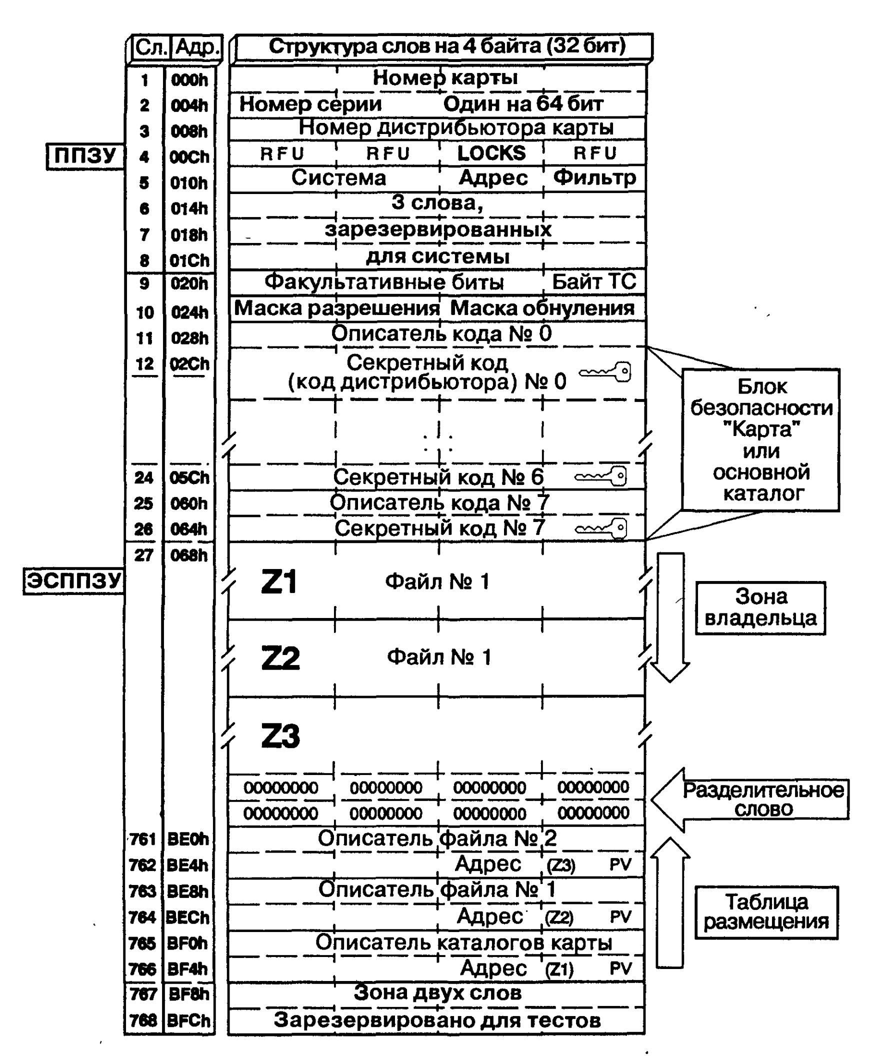


Рис. 1.13. Картография памяти карт COS на 24 Кбайт ЭСППЗУ

демонстрирует картографию модели на 24 Кбайт; оба варианта нашли исключительно широкое применение.

Хотя в случае необходимости каждый байт памяти может быть прочитан или записан напрямую, принцип COS позволяет организовать карту в виде некоторого числа файлов, по аналогии с дискетами в DOS (Disk Operating System – сходство очевидно).

Естественно, в каждом файле могут использоваться многочисленные системы безопасности, указанные в соответствующем *описателе*, или *дескрипторе*. В частности, операции записи и/или считывания могут проводиться по представлении конфиденциальных кодов (каждая карта поддерживает до 7 различных кодов).

Некоторые карты COS снабжены алгоритмом кодирования DES, позволяющим шифровать обмен данных между картой и считывающим устройством: например, не существует возможности перехватить конфиденциальный код, представленный карте.

С точки зрения технологии память подразделяется на две смежные области: зону ППЗУ на 32 байта, расположенную в области младших адресов, где возможна лишь простая запись, – данные из этой части никогда не могут быть стерты или открыты, начиная с индивидуального серийного номера каждой карты, – и вторую зону, которая включает в себя остальную память типа ЭСППЗУ. Здесь данные могут быть стерты и переписаны по желанию.

Все пространство памяти состоит из слов по 32 бита, то есть по 4 байта. В зоне ППЗУ расположена прежде всего область идентификации карты (номер серии, записанный изготовителем) и дистрибьютора (идентифицирующая запись, выполненная организацией, которая распространяет карты). Речь идет о словах от 1 до 3. Следующая область представляет собой *зону контроля кода* ПЗУ. Расположенная между зонами ППЗУ и ЭСППЗУ, она содержит 6 слов (от 4 до 9). Слово 4 включает в себя *замки* (locks) карты, в частности биты BFAB и BPERS, представленные на рис. 1.14, установление которых в 1 является необратимым; в процессе изготовления оба бита BFAB и BPERS на карте находятся в нуле, а слово 4 будет, например, 00h. На данный момент карта абсолютно не защищена.

На любой карте, покидающей завод (снабженной серийным номером, но еще не *персонализированной* дистрибьютором), BFAB равен 1, а BPERS – 0: слово 4 имеет значение 04h. В частности, так устроены карты-образцы.

Когда дистрибьютор заканчивает свою работу по персонализации, он выставляет бит BPERS в 1. Слово 4 будет тогда 0Ch – это значение можно встретить с наибольшей вероятностью, когда работа ведется с картами, выпущенными в открытое обращение; слово 5, называемое *фильтром*, содержит точку входа в прикладную программу, в случае необходимости записываемую в память ЭСППЗУ. Если оба последних байта слова 5 равны 00h, выполняется стандартная программа COS.

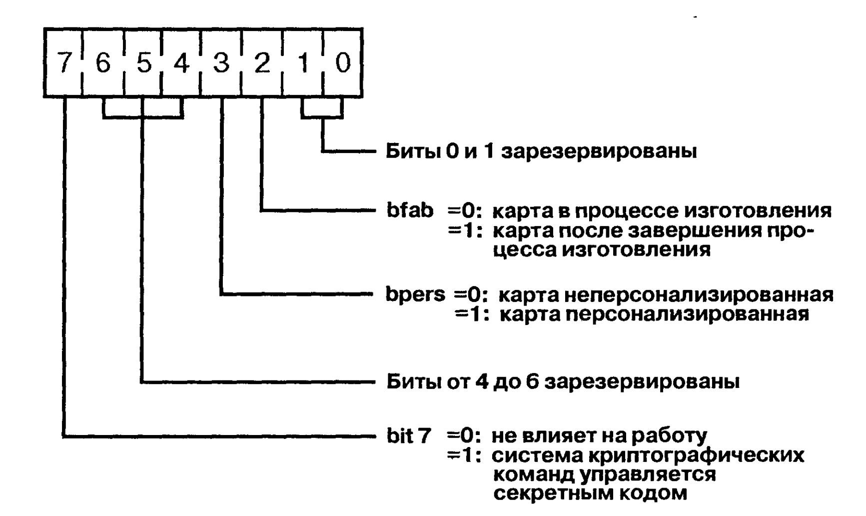


Рис. 1.14. Различные «замки» карт семейства COS

Здесь речь идет о чрезвычайно мощной способности COS, позволяющей разработчику обусловить «поведение» карты, почти всецело зависящее от программы, которую он напишет, базируясь на системе команд микропроцессора (обычно ST16XYZ SGS-Thomson). Слова 6 и 8 зарезервированы для системы; слово 9 содержит все опции, которые будут определять работу базовой программы в ПЗУ: тип протокола передачи, режим размещения файлов, класс применения, некоторые элементы ответа на *сброс* (reset) и т.д.

Затем, начиная со слова 10 (адрес 024h), идет *блок безопасности* карты, включающий, в частности, конфиденциальные коды и их описатели (дескрипторы). Сразу за блоком безопасности следуют файлы, контролируемые *таблицей размещения*, которая содержится в конце того же пространства памяти и которую можно сравнить с FAT дискеты. Наконец, два *тестовых слова* расположены в самой верхней области памяти: 64 бита, которые можно свободно читать и записывать.

ГЛАВА СТРАНИЦА

2 ИССЛЕДОВАНИЯ БАНКОВСКОЙ КАРТЫ

При отсутствии карты с истекшим сроком службы нижеописанные операции с тем же успехом можно применить к еще действующим картам, так как в основном речь пойдет об операциях чтения. Необходимо иметь в виду, что карты France Telecom (ранее они назывались PASTEL) очень близки к банковским и также могут служить исходным материалом.

Важно отметить, что исследователь целиком и полностью берет на себя ответственность за свои опыты.

ВАРИАНТ МАСКИ BULL CP8 M4

Перед тем как приступить к чтению банковских карт, важно вспомнить, что классом ISO карт BULL CP8 является BCh; регистр-указатель ADL имеет значение 08E0h для всех французских банковских карт, а также для карт France Telecom. Этого достаточно, чтобы с помощью ПК, снабженного соответствующим устройством чтения-записи, изучить данные, которые можно свободно прочитать. Но, начиная с момента, когда в распоряжение пользователя поступает конфиденциальный код испытуемой карты (что вполне законно, если экспериментатор является ее владельцем), возникает искушение вскрыть некоторые зоны, защищенные более надежно.

По умолчанию, если не дано обратное условие, ADC равно ADT (то есть не существует собственно «конфиденциальной зоны»), и эти слова могут принимать значения 02E0h, 0280h, 02B0h, 0360h и т.д. ADM имеет адрес 0260h на «устаревших» банковских картах с памятью ППЗУ (так называемая маска B0). На новых картах с памятью ЭСППЗУ (так называемая маска B0') ADM, скорее всего, имеет значение 0290h. В связи с этим «зона доступа» (называемая здесь «зоной состояния») существенно меньшего размера.

Это логическое следствие изменения технологии, позволяющее повторно использовать данную зону, полностью исключает риск переполнения, на которое в прежние времена жаловались владельцы карт с большим кредитом.

Легко отличить карты B0 от B0', учитывая, что у последних обычно только один чип в ориентации ISO, называемом «центральным», и шесть контактов вместо восьми. Но даже в случае обладания правильным конфиденциальным кодом не стоит надеяться проникнуть в секретную зону. При подобных попытках карта просто перестанет реагировать на ваши действия. Само собой разумеется, если бы вас ждало нечто иное, эта глава, а может быть, и вся книга не были бы написаны.

Безусловно, информация, которой располагает автор, не может считаться точной.

Учитывая возможную унификацию на мировом рынке кредитных чип-карт, следует ожидать серьезных изменений как в аппаратном, так и в программном обеспечении. Предвестником этого процесса можно считать спецификацию EMV (Europay, Mastercard, Visa), из которой следует, например, что у карт для ведения денежных расчетов классом ISO может стать 80h. Время покажет, насколько справедливы эти прогнозы.

УСТРОЙСТВО ЧТЕНИЯ-ЗАПИСИ ДЛЯ КАРТ НА МИКРОПРОЦЕССОРЕ

Проведение экспериментов с банковскими картами (а в широком смысле – со всеми картами CP8) существенно облегчается благодаря набору для самостоятельного изучения, разработанному концерном BULL и выпущенному на рынок компанией TEKELEC. В комплект входят устройство чтения-записи TLP224, двадцать карт SCOT 50, серия справочников и целая библиотека программного обеспечения. С помощью этого набора можно научиться эффективно работать с данным семейством карт. Также здесь содержатся все необходимые фундаментальные элементы для ведения разработок прототипов настоящих приложений по всем правилам искусства.

Представляемая документация основана на «Руководстве по применению семейства SCOT», объемном справочном издании, содержащем всю информацию о данных картах. Его изучение чрезвычайно пригодилось автору для успешного ведения исследований, результаты которых представлены в данной книге.

Однако наибольший интерес представляет программное обеспечение из указанного набора. Параллельно с несколькими программами, носящими скорее демонстративный характер, следует по достоинству оценить драйверы и библиотеки на языке С, которые в значительной степени облегчат разработку реальных приложений.

Метод, которому отдается предпочтение в данной книге, основан на самостоятельном изготовлении устройства чтения-записи из легко доступных элементов.

Речь идет об устройстве, подробно описанном в книге «Чип-карты. Устройство и применение в практических конструкциях». Все программное обеспечение для карт на микропроцессоре было написано специально для этого устройства и представлено в указанной

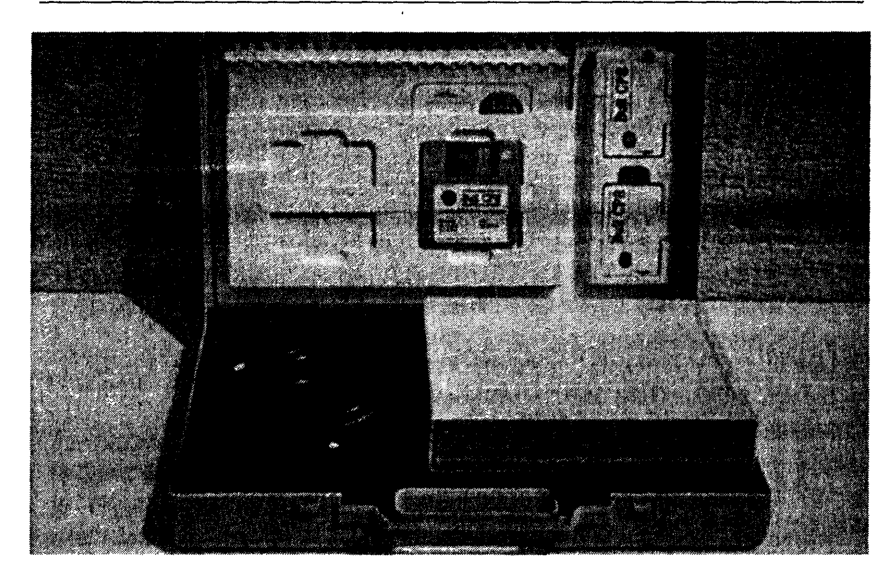


Рис. 2.1. Набор для самостоятельного изучения чип-карт BULL CP8

книге, а исходные тексты русских версий программ размещены на сайте www.dmk.ru. При необходимости можно модернизировать его и адаптировать для другого применения.

В данной книге воспроизводится схема устройства чтения-записи, построенного на базе микроконтроллера PIC 16C84, который запрограммирован «прошивкой» из файла COUP84.HEX, представленного на сайте www.dmk.ru. Микроконтроллер PIC 16C84 играет роль упрощенного блока сопряжения.

Операция программирования может выполняться с помощью простейшего программатора для микроконтроллера PIC, описанного в книге “Composants electroniques programmables sur PC”, вышедшей во Франции[1].

Если принимается решение использовать другое программирующее устройство (например, PICSTART 16B производства компании Microchip), то «шить» следует данные из файла COUP84.OBJ. Кроме этого, нужно вручную выставить перемычки конфигурации микроконтроллера PIC: биты генератора – в режим XT, биты WDT и PWRT – в положение «ВКЛ», а бит защиты кода от копирования – по желанию конструктора.

[1] Русский перевод: «Как превратить персональный компьютер в универсальный программатор», М.: ДМК, 2000.

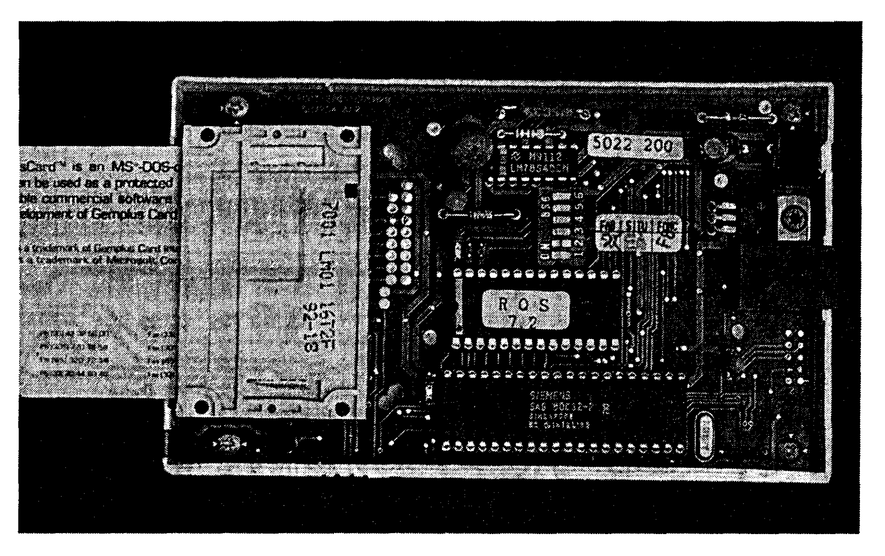


Рис. 2.2. Устройство чтения-записи GCR200 Gemplus Card International

Рис. 2.3 воспроизводит топологию печатной платы, а рис. 2.4 – схему размещения элементов. Принципиальная схема и исходный текст используемого программного обеспечения не являются необходимыми для сборки, поэтому здесь не представлены.

Внимание! Разъем DB9 (соединение по протоколу RS232 с портом COM1 ПК) должен быть обязательно типа «розетка», а кабель – типа «удлинитель» (вилка-розетка). Разъем типа «вилка» в сочетании с кабелем «две розетки» не подойдет.

Схема требует внешнего питания 5 В, в случае необходимости напряжение программирования Vpp должно подаваться на второй клеммник, но только при проведении некоторых специальных операций, причем с определенными типами карт, изготовленных по старой технологии.

Теперь остается создать блок картоприемника, который будет соединен со схемой устройства чтения-записи коротким плоским кабелем, снабженным двумя одинаково смонтированными разъемами HE10 по 10 контактов. На рис. 2.5 показана топология печатной платы, а на рис. 2.6 – соответствующая схема размещения элементов.

Отметим, что для разъемов HE10 предусмотрены два приспособления (в данном случае две разрезные колодки из двух рядов квадратных штырьков): первое – для чип-карт стандарта ISO (центральное расположение контактов), а второе – для стандарта AFNOR

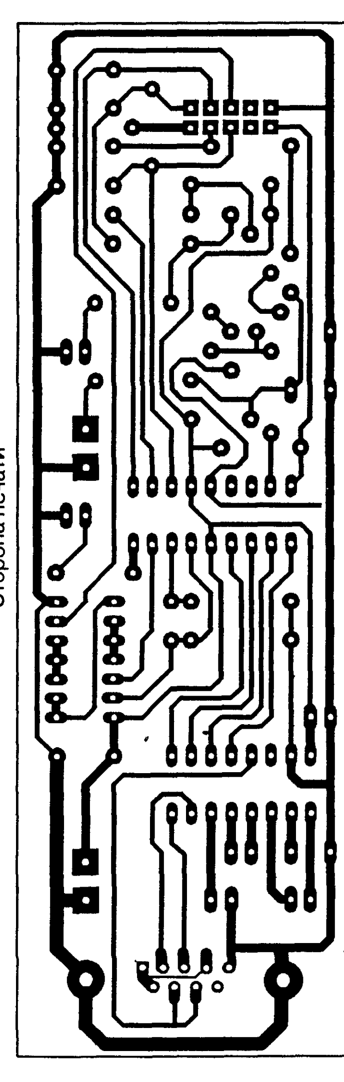


Рис. 2.3. Печатная плата универсального устройства чтения-записи

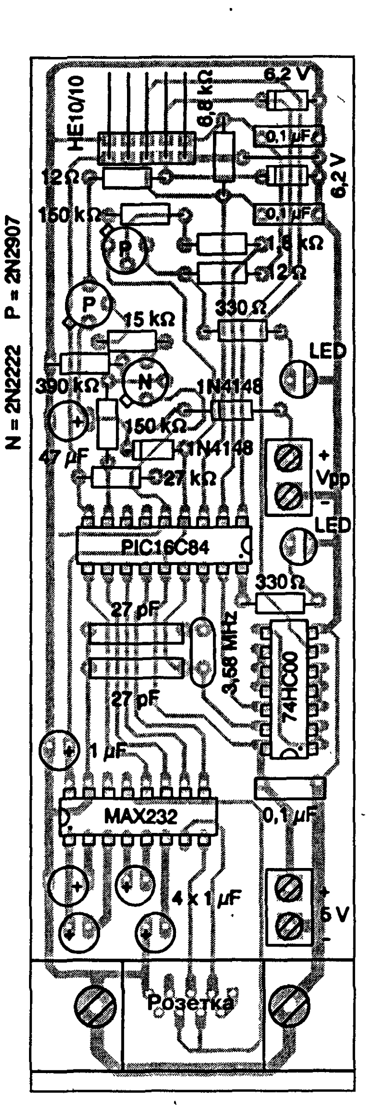


Рис. 2.4. Схема размещения элементов на плате универсального устройства чтения-записи

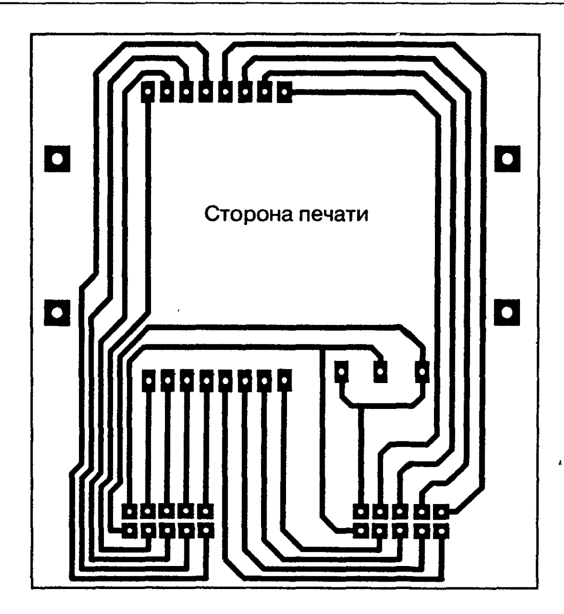


Рис. 2.5. Печатная плата картоприемника ISO/AFNOR

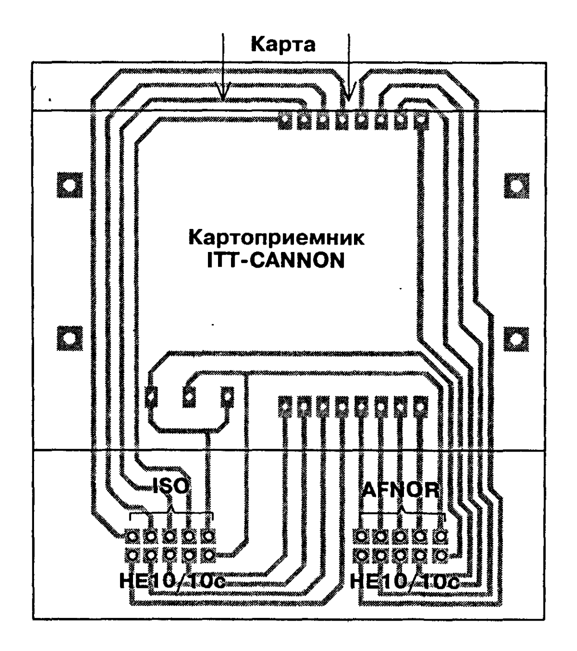


Рис. 2.6. Схема размещения элементов блока картоприемника

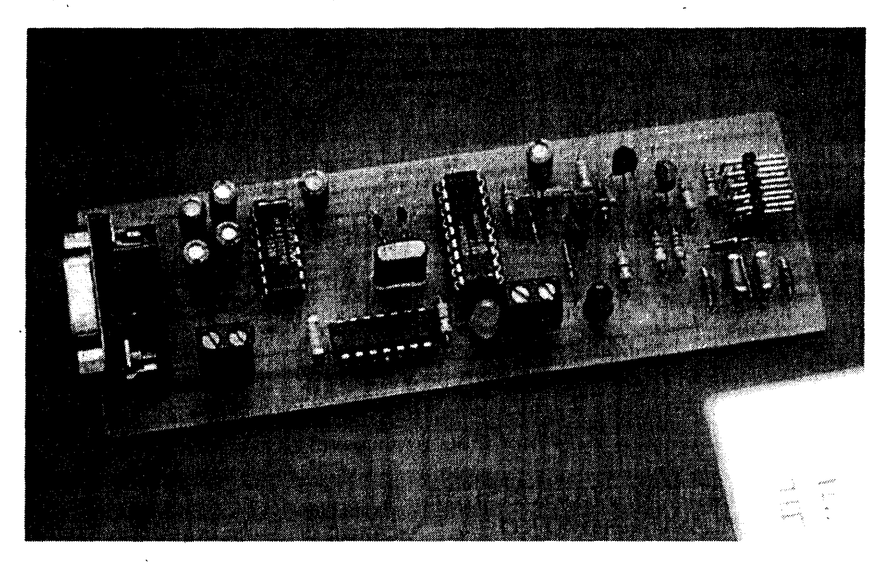


Рис. 2.7. Универсальное устройство чтения-записи

(эксцентричное, «смещенное» расположение контактов). Естественно, для каждого типа карт необходимо применять соответствующий *картоприемник.*

Конечно, можно заняться исследованиями банковских карт, используя программу, способную вести диалог в соответствии с нормами ISO 7816.

Поскольку в настоящий момент банковские карты работают по так называемому *обратному соглашению* и имеют протокол T=0, предлагается использовать программу INVERSE.BAS, представленную в вышеупомянутом издании, посвященном чип-картам. Ее исполняемая версия INVERSE.EXE находится на сайте www.dmk.ru. Для карт, работающих по *прямому соглашению*, необходимо использовать соответствующую программу DIRECTE.EXE, которая позволяет отправлять на карту любую команду (как на ввод, так и на вывод) согласно стандартам ISO и показывать ответ. Напомним, что команда на ввод направляет блок данных на карту, а на вывод – принимает данные от нее.

Имея в распоряжении систему команд, распознаваемую банковскими картами (табл. 2.1), можно начать исследование любой карты такого рода.

Логично будет начать исследования с *зоны чтения.* Ее примерная картография представлена в табл. 2.2. Нужно заметить, что любое

слово из 32 бит начинается с *контрольного ключа*; он, как правило, равен 0011, но установлен на 0010, если отмечает начало так называемого *представительского блока*.

Наиболее интересен участок № 2, содержащий данные, во многом эквивалентные тем, которые оттиснуты («выдавлены») на пластиковом материале карты и записаны на магнитных полосах: номер карты, личность ее владельца, сроки действия и т.д. Участок № 3 («происхождение подтверждено») играет защитную роль: он содержит «значение аутентификации», закодированное в 320 битах.

Скорее всего, именно результат криптографического вычисления позволяет задействовать содержимое участка 02 и «секретный набор», который невозможно прочитать напрямую ни при каких условиях. Это один из способов удостовериться в подлинности карты.

ПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ КОНФИДЕНЦИАЛЬНОГО КОДА

При попытке проведения операции чтения по адресам, располагающимся ниже ADL, карте необходимо предъявить конфиденциальный код ее владельца.

Таблица 2.1. Система команд для банковских карт M4 B0'

Команда	Порядок выполнения команд
0E	Стирание
10	Представление банковского ключа (CB)
20	Представление конфиденциального кода (CC) или кода разблокировки карты
30	Представление кода открытия (CO)
40	Утверждение операции чтения
50	Запись замка
70	Утверждение операции записи
80	Сертификация с помощью первой секретной комбинации
82	второй секретной комбинации
84	третьей секретной комбинации
86	четвертой секретной комбинации
88	пятой секретной комбинации
A0	Поиск первого незаполненного слова или слова по аргументу
A8	Поиск первого заполненного слова
B0	Считывание байтов
C0	Считывание результатов
D0	Запись слова

Таблица 2.2. Картография зоны считывания банковских карт

ADL (08E0)	0010	1110	Участок 03	L=48	Ключ		CCE
08E8	0011	0000	0000	0000	0000		
08F0	0011	320 бит (величина аутентификации)					
***							
(0948)	0010	1110	Участок 02	L=56	111		CCE
	0011	Код записи 00	№ карты (19 знаков BCD)				
***							
0968	0011	Использование (3 знака BCD)	Начало срока действия (4 знака BCD)				
0970	0011	Язык (3 знака BCD)	Окончание срока действия (4 знака BCD)				
0978	0011	Валюта (3 знака BCD)	Порядок числа	Двоичный номер (начало)			
0980	0011	Личность владельца (26 знаков ASCII)					
***							
09B8	0011	Личность владельца	RUF	Двоичный номер (продолжение)			
09C0	00	AD1	CCE	RUF			
09C8	00	ADL	CCE	ADT			CCE
09D0	00	ADC	CCE	ADM			CCE
09D8	00	AD2	CCE	ADS			CCE
09E0	Тип = 3FE5 (CB)						CCE
09E8	00	AD1	CCE	№ изготовителя			CCE
0 F0	№ серии						CCE
09F8				10011			

На этом этапе выявляется особенность, присущая картам CP8, которые требуют, чтобы любой предъявленный код был затем подтвержден (или ратифицирован) с помощью специальной системы. Следовательно, будет недостаточно отправить команду представления кода формы BC 20 00 00 04, а потом четырех байт PIN-кода.

Для введения в действие системы ратификации кода (40h) по адресу 00h с длиной данных 00h необходимо также отправить следующую строку: BC 40 00 00 00. Но это еще не все! На самом деле обычный конфиденциальный код состоит из четырех цифр, в то время как карта ждет слова из четырех байт (восемь шестнадцатеричных цифр). Код 4950, например, должен быть представлен в шестнадцатеричном формате 12 54 3F FF. Таким образом, необходимо «транскодировать» конфиденциальные коды карт, в данном случае при помощи небольшой программы PIN2CB.BAS:

```
10 REM -- PIN2CB.BAS --
20 KEY OFF :CLS :INPUT "PIN: ",P$:K$=""
30 FOR F=1 TO LEN(P$)
40 M$="&h"+MID$(P$,F,1)
50 M=VAL(M$)
60 D=0
70 IF M>7 THEN D=1 :M=M-8
80 GOSUB 280
90 IF M>3 THEN D=1 :M=M-4
100 GOSUB 280
110 IF M>1 THEN D=1 :M=M-2
120 GOSUB 280
130 D=M :GOSUB 280
140 NEXT F
150 K$="00"+K$+"11111111111111"
160 PRINT :PRINT"Представить карте шестизначный код: ";
170 FOR F=1 TO LEN(K$) STEP 4
180 D$=MID$(K$,F,4)
190 A=0
200 FOR G=1 TO 4
210 B$=MID$(D$,5-G,1)
220 IF B$="1" THEN A=A+2^(G-1)
230 NEXT G
240 A$=HEX$(A)
250 PRINT A$;
260 NEXT F :PRINT :PRINT
270 END
280 IF D=0 THEN K$=K$+"0"
290 IF D=1 THEN K$=K$+"1"
300 D=0 :RETURN
310 REM (c)1994 Patrick GUEULLE
```

Разблокировка для считывания рабочей зоны потребует выполнения следующих операций:

1. Запуск программы INVERSE.EXE, затем размещение карты в картоприемнике.
2. Ожидание ответа на сброс от карты.
3. Запуск команды представления PIN-кода (BC 20 00 00 04).
4. Ожидание ответа карты (байт операции 20h).
5. Отсылка напрямую четырех байт кода (в нашем примере 12 54 3F FF).
6. Ожидание протокола-подтверждения правильного выполнения (если все нормально, то 90 00; появление значения отличного от 00 может свидетельствовать о том, что во время предыдущей сессии был представлен и ратифицирован ложный код).

7. Ратификация кода (BC 40 00 00 00).
8. Ожидание байта операции (40h или 41h), за которым сразу следует протокол правильного выполнения (90 00, если был представлен правильный код).

Отметим, что байт операции равный 41h вместо 40h говорит о том, что карта требует напряжения программирования Vpp. Если это не выполняется, в протоколе появляется значение отличное от 90 00, но представленный ранее код не считается ложным.

В таком случае надо всю процедуру начать заново. При этом напряжение программирования Vpp (часто оно составляет 21 В, однако необходимо проверять его в каждом конкретном случае) подается непосредственно перед началом ратификации, после чего следует его снять, так как операция записи не предусматривается.

Теперь остается только перейти к считыванию блока байтов внутри этой зоны столько раз, сколько необходимо. Зона начинается с адреса 0290h или 0260h (в зависимости от возраста карты).

Команда на ввод в карту будет иметь форму DC DO WX YZ LL, где приняты следующие обозначения: WXYZ – адрес начала зоны считывания (в шестнадцатеричной системе счисления), LL – число байтов для считывания (тоже в шестнадцатеричной системе). Чтобы прочитать 32 байта начиная с адреса 0290h, отправляют соответственно команду в форме BC B0 02 90 20.

Блок байтов, полученный в ответ, поступит в сопровождении «байта процедуры» B0h и протокола, который, если все прошло нормально, будет иметь значение 90 00. Обычно там должна располагаться последовательность нулей, отделенная от последовательности F цифрой, которая может равняться 1, 3 или 7 (0001, 0011 или 0111 в двоичной системе счисления).

По информации автора, каждое представление с ратификацией правильного конфиденциального кода приводит к обнулению бита из указанной последовательности битов равных 1. Таким образом, на кристалле находится настоящий «счетчик», наблюдающий за работой карты. Он может оказаться «переполненным» (для карт B0), если использование было слишком интенсивным. Конечно, есть возможность проверить, не использовалась ли карта без ведома ее владельца, но – обратите внимание! – при процедуре контроля, запрашивающего представление конфиденциального кода, всегда «съедается» один бит.

Наконец, создается впечатление, что представление ложного кода с проведенной ратификацией оставляет след в виде единицы, не трансформированной в ноль.

КОНТРОЛЬ РАСХОДОВ

Последовательные блоки из четырех байт представляют хронологию расчетов, производимых месяц за месяцем с помощью чип-карты (и ни в коем случае не магнитных полос).

Посредством раскодирования слов, состоящих из 32 бит, разыскиваются «следы» каждой покупки (с точностью до сантима или франка) и даты ее приобретения. При желании можно даже создать настоящую «распечатку счетов». Например, слово 30 00 95 12 могло бы обозначать начало декабря 1995 года, а слова 33 58 38 18 или 33 98 4C F4, стоящие сразу после него, выявили бы две покупки, сделанные тогда же по представлении конфиденциального кода чипу.

Несомненно, ведение на самой карте такой детальной хронологии всех коммерческих операций объясняет феномен переполнения, сигнал о котором иногда появляется до истечения двухгодичного срока действия банковских чип-карт. Это вызвано тем, что маска B0 использует память ППЗУ, где исключено перепрограммирование зон.

Новые карты B0', выполненные по технологии ЭСППЗУ (с возможностью повторной записи), подвергаются процедуре, при которой давние коммерческие операции стираются, чтобы уступить место более поздним.

НАБОР РАБОЧИХ ИНСТРУМЕНТОВ ДЛЯ БАНКОВСКОЙ КАРТЫ

Хотя программа INVERSE.EXE позволяет выполнять любые операции с банковскими картами, ее применение может оказаться трудоемким, так как группы байтов, получаемые с карты, требуют особого раскодирования. В особенности это проявляется при помещении каких-либо данных в зону чтения и знаменитой «распечатке счетов», о существовании которой говорилось чуть выше.

Автором были разработаны специальные программы раскодирования наиболее интересных данных. Программы чтения и раскодирования ответа на сброс (ATR.EXE и DECATR.EXE), по сути, являются частями упомянутого набора инструментов. Они представлены на дискете, прилагаемой к французкому изданию книги “Cartes à puce, Initiation et applications”, а российские версии программ размещены на сайте www.dmk.ru. В настоящем издании они не приводятся, поскольку для выполнения нижеописанных манипуляций не понадобятся.

```
10 REM -- ADL.BAS --
20 KEY OFF:CLS
30 PRINT"ADL (c)1997 Патрик Гелль. Чтение открытой зоны банк. карты B0 или B0'"
```

```
40 PRINT :PRINT
50 OPEN "COM1:9600,o,8,2" AS #1
60 PRINT"Вставить карту в картоприемник"
70 IF LOC(1)<>0 THEN 100
80 IF INKEY$=CHR$(27) THEN END
90 GOTO 70
100 T=TIMER :BEEP
110 IF T>TIMER-1 THEN 110
120 A$=""
130 WHILE NOT EOF(1)
140 A$=A$+INPUT$(1,#1)
150 WEND
160 IF LEFT$(A$,1) = CHR$(3) THEN 180
170 CLS :PRINT"Карта не опознана" :END
180 B$="BCB008E090"
190 PRINT :PRINT"Команда отправлена на карту: ",200 FOR F=1 TO 9 STEP 2
210 K$=MID$(B$,F,2)
220 PRINT K$;" "; :N$="&h"+K$
230 N=VAL(N$) :GOSUB 620
240 PRINT#1,CHR$(M);
245 FOR T=0 TO 100 :NEXT T
250 NEXT F
260 PRINT"(ADL=08E0h)"
270 PRINT :PRINT"Ответ карты:" :PRINT
280 T=TIMER
285 IF T>TIMER-1 THEN 285
290 IF LOC(1)<>0 THEN 320
300 IF INKEY$=CHR$(27) THEN END
310 GOTO 290
320 A$=INPUT$(LOC(1),#1)
330 D$=""
340 FOR F=1 TO LEN(A$)
350 N$=MID$(A$,F,1)
360 N=ASC(N$)
370 GOSUB 620
380 M$=HEX$(M)
390 IF LEN(M$)=1 THEN M$="0"+M$
400 PRINT M$+" "; :D$=D$+M$
410 NEXT F
420 IF LOC(1)<>0 THEN A$=INPUT$(LOC(1),#1):GOTO 340
430 CLOSE#1
440 PRINT :PRINT :PRINT"Создание файла ADL.CAR ";
450 OPEN "ADL.CAR" FOR OUTPUT AS #1
460 K=0 :PRINT
470 FOR F=3 TO LEN(D$)-5 STEP 2
480 B$="&h"+MID$(D$,F,2)
490 B=VAL(B$)
500 IF(B AND 128=128 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
510 IF(B AND 64)=64 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
520 IF(B AND 32)=32 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
```

```
530 IF(B AND 16)=16 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
540 IF(B AND 8)=8 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
550 IF(B AND 4)=4 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
560 IF(B AND 2)=2 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
570 IF(B AND 1)=1 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
580 K=K+1
590 IF K>3 THEN PRINT#1,:K=0
600 NEXT F
610 END
620 M=255
630 IF N>127 THEN N=N-128:M=M-1
640 IF N>63 THEN N=N-64:M=M-2
650 IF N>31 THEN N=N-32:M=M-4
660 IF N>15 THEN N=N-16:M=M-8
670 IF N>7 THEN N=N-8:M=M-16
680 IF N>3 THEN N=N-4:M=M-32
690 IF N>1 THEN N=N-2:M=M-64
700 IF N>0 THEN M=M-128
710 RETURN
720 REM (c)1997 Patrick GUEULLE
```

Программа ADL.BAS должна рассматриваться как исходный текст ADL.EXE – компилированного исполняемого файла, который должен запускаться из командной строки DOS или, при необходимости, в диалоговом окне Windows 95. Это вызвано конфигурацией последовательного порта, из-за которой интерпретирующие программы типа GWBASIC «не понимают» некоторых опций, необходимых для нормальной работы конструкций, рассматриваемых в книге.

Такие программы просто считывают информацию зоны чтения любой совместимой карты (банковской или телефонной) и сохраняют ее в бинарном виде в текстовом файле ADL.CAR. Раскодирование этих данных затем будет доверено программе DECADL.BAS, откомпилированная версия которой, DECADL.EXE, также представлена на сайте www.dmk.ru:

```
10 REM -- DECADL.BAS --
20 OPEN "ADL.CAR" FOR INPUT AS#1
30 KEY OFF :GOSUB 940
40 IF C$=CHR$(46) THEN 60
50 CLS :PRINT"Этот файл не соответствует банковской карте" :BEEP :END
60 CLS :PRINT"DECADL (c)1997 Патрик Гелль. Анализ файла ADL.CAR"
70 FOR F=1 TO 452 :INPUT#1,N :NEXT F
80 PRINT :PRINT :PRINT"Карта номер: ";
90 FOR H=1 TO 5 :GOSUB 860 :NEXT H
100 GOSUB 1030
110 FOR H=1 TO 7 :GOSUB 860 :NEXT H
120 GOSUB 1030
```

```
130 FOR H=1 TO 7 :GOSUB 860 :NEXT H
140 GOSUB 1030
150 PRINT :PRINT"Код услуги: ";
160 FOR H=1 TO 3 :GOSUB 860 :NEXT H
170 PRINT :PRINT"Начало срока действия: ";
180 FOR H=1 TO 4 :GOSUB 860 :NEXT H
190 GOSUB 1030
200 PRINT :PRINT"Код языка: ";
210 FOR H=1 TO 3 :GOSUB 860 :NEXT H
220 PRINT" (250 = французский)";
230 PRINT"Конец срока действия,:";
240 FOR H=1 TO 4 :GOSUB 860 :NEXT H
250 GOSUB 1030
260 PRINT :PRINT"Код валюты: ";
270 FOR H=1 TO 3 :GOSUB 860 :NEXT H
280 PRINT" (250 = французские франки)";
290 PRINT"Множитель: ";
300 GOSUB 860
310 PRINT" (3 = сотые, 5 = единицы)";
320 FOR H=1 TO 4 :GOSUB 1030 :NEXT H
330 PRINT"Владелец: ";
340 FOR J=1 TO 3
350 FOR H=1 TO 3 :GOSUB 940 :NEXT H
360 GOSUB 1080
370 FOR H=1 TO 3 :GOSUB 940 :NEXT H
380 GOSUB 1030
390 NEXT J
400 FOR H=1 TO 3 :GOSUB 940 :NEXT H
410 GOSUB 1080 :GOSUB 940
420 FOR H=1 TO 8 :GOSUB 1030 :NEXT H
430 PRINT :PRINT :PRINT"Слово опции: ";
440 FOR H=1 TO 11
450 INPUT#1,N :PRINT N;
460 NEXT H
470 FOR H=1 TO 5 :INPUT#1,N :NEXT H
480 PRINT :PRINT"ADL = "; :GOSUB 1220
490 FOR H=1 TO 5 :INPUT#1,N :NEXT H
500 PRINT" "; :PRINT"ADT = "; :GOSUB 1220
510 FOR H=1 TO 5 :INPUT#1,N :NEXT H
520 PRINT" "; :PRINT"ADC = "; :GOSUB 1220
530 FOR H=1 TO 5 :INPUT#1,N :NEXT H
540 PRINT" "; :PRINT"ADM = "; :GOSUB 1220
550 FOR H=1 TO 5 :INPUT#1,N :NEXT H
560 PRINT" "; :PRINT"AD2 = "; :GOSUB 1220
570 FOR H=1 TO 5 :INPUT#1,N :NEXT H
580 PRINT" "; :PRINT"ADS = "; :GOSUB 1220
590 FOR H=1 TO 29 :INPUT#1,N :NEXT H
600 PRINT :PRINT"Ep = ";N,"(Зона коммерческой операции ";
610 IF N=1 THEN PRINT"нет)";
620 PRINT"защита от записи)"
630 PRINT"Lp = ";
```

```
640 INPUT#1,N
650 PRINT N,"(Зона коммерческой операции ";
660 IF N=1 THEN PRINT"нет)";
670 PRINT"защита от чтения)";
680 FOR H=1 TO 7 :INPUT#1,N :NEXT H
690 PRINT"AD1 = "; :GOSUB 1220
700 FOR H=1 TO 12 :INPUT#1,N :NEXT H
710 PRINT :PRINT"Номер изготовителя: ";
720 GOSUB 860
730 IF C=1 THEN PRINT" (CP8 Oberthur)"
740 IF C=2 THEN PRINT" (Philips TRT)"
750 IF C=3 THEN PRINT" (Gemplus)"
760 IF C=4 THEN PRINT" (Solaic)"
770 IF C=5 THEN PRINT" (Schlumberger)"
780 FOR H=1 TO 6 :INPUT#1,N :NEXT H
790 PRINT"Номер серии: ";
800 FOR H=1 TO 26 :INPUT#1,N :PRINT N; :NEXT H
810 FOR H=1 TO 21 :INPUT#1,N :NEXT H
820 PRINT: PRINT"Слово замков: ";
830 FOR H=1 TO 5 :INPUT#1,N :PRINT N; :NEXT H
840 PRINT" (10011 = карта кредитоспособна)"
850 PRINT :END
860 REM - BCD -
870 C=0
880 FOR G=0 TO 3
890 INPUT#1,N
900 IF N=1 THEN C=C+2^(3-G)
910 NEXT G
920 IF C<=9 THEN PRINT C;
930 RETURN
940 REM - ASCII -
950 C=0
960 FOR G=0 TO 7
970 INPUT#1,N
980 IF N=1 THEN C=C+2^(7-G)
990 NEXT G
1000 C$=CHR$(C)
1010 PRINT C$;
1020 RETURN
1030 REM - Пропустить четыре строки -
1040 FOR F=1 TO 4
1050 INPUT#1,N
1060 NEXT F
1070 RETURN
1080 REM -- ASCII с разрядкой, --
1090 C=0
1100 FOR G=0 TO 3
1110 INPUT#1,N
1120 IF N=1 THEN C=C+2^(7-G)
1130 NEXT G
1140 GOSUB 1030
```

```
1150 FOR G=0 TO 3
1160 INPUT#1,N
1170 IF N=1 THEN C=C+2^(3-G)
1180 NEXT G
1190 C$=CHR$(C)
1200 PRINT C$;
1210 RETURN
1220 REM - BIN -
1230 C=0
1240 FOR G=0 TO 10
1250 INPUT#1,N
1260 IF N=1 THEN C=C+2^(10-G)
1270 NEXT G
1280 C$=HEX$(C*8)
1290 IF LEN(C$)<4 THEN C$="0"+C$
1300 PRINT C$;
1310 RETURN
1320 REM (c)1997 Patrick GUEULLE
```

Ниже показан результат работы программы (при этом изъяты некоторые персональные сведения).

```
DECADL (c)1997 Patrick GUEULLE Анализ файла ADL.CAR
Карта номер: 4 9 7 xxxxxxxxxxxxx
Код услуги: 1 0 1
Начало срока действия: 9 5 1 1
Код языка: 2 5 0 (250 = французский)
Конец срока действия: 9 7 1 1
Код валюты: 2 5 0 (250 = французские франки)
Множитель: 3 (3 = сотые, 5 = единицы)
Владелец: MR xxxxxxxxxxxxxxxxxxx
Слово опции: 1 1 1 1 0 1 0 1 0 0 0
ADL = 08E0 ADT = 02B0 ADC = 02B0 ADM = 0290 AD2 = 0278 ADS = 0230
Ep = 0 (Зона коммерческой операции защита от записи)
Lp = 0 (Зона коммерческой операции защита от чтения)
AD1 = 0210
Номер изготовителя: 1 (CP8 Oberthur)
Номер серии:
0 0 0 1 0 1 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1
Слово замков: 1 0 0 1 1(10011 = карта кредитоспособна)
```

Программа ADT.EXE (откомпилированная версия ADT.BAS) позволяет пойти значительно дальше. С ее помощью можно получить доступ к рабочей зоне, которая, как известно, защищена посредством конфиденциального или PIN-кода.

Прежде чем создать файл ADT.CAR (он имеет такой же формат, как и ADL.CAR), программа просит владельца карты сообщить свой конфиденциальный код из четырех цифр (транскодирование на четыре байта будет произведено автоматически), а также запрашивает начало рабочей зоны (ADT). Последняя информация предоставляется DECADL.BAS.

Таким образом считываются 252 байта, которых вполне достаточно, и результаты выводятся на экран в следующем виде:

```
Команда отправлена на карту: BC 40 00 00 00
Ответ карты:
40 90 00
Чтение зоны транзакций
Команда отправлена на карту: BC B0 02 B0 FC
Ответ карты:
B0 30 00 95 11 B0 00 95 12 30 00 95 12 33 D0 2B AC 30 00 96 01 33 10 6B D0 33
38 76 C0 30 00 96 02 33 68 3A 50 33 80 3C F0 33 C0 2A BC 33 D1 0C 48 33 D1 0C
48 33 D1 0C 48 30 00 96 03 33 48 2E D6 33 58 47 E9 33 E9 62 9B 33 F8 43 1A 30
00 96 04 33 28 3F 9C 33 E8 38 E0 30 00 96 05 33 A8 37 EC 30 00 96 06 33 C8 4E
20 30 00 96 07 33 10 37 14 33 43 36 41 33 68 48 A8 33 78 2D 50 33 78 24 C2 33
A0 2F F8 33 E8 3C 5F FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 90 00
Создание файла ADT.HEX
```

Теперь достаточно запустить DECADT.BAS (или DECADT.EXE), чтобы раскодировать «распечатку счетов», записанную в ADT.CAR. Для этого программа обращается к «порядку числа» карты – цифровому значению, предоставленному опять же DECADL.BAS, которое уточняет, в каких денежных единицах указаны суммы: франках или сантимах.

Ниже воспроизводится отрывок из такой «распечатки счетов» с точностью до сантима, так как она была прочитана на карте, порядок содержимого которой равен 3:

```
 7/01/96 : 304.00 F
13/02/96 : 149.28 F
16/02/96 : 156.00 F
24/02/96 : 109.40 F
26/02/96 : 686.80 F
 9/03/96 : 119.90 F
11/03/96 : 184.09 F
29/03/96 : 907.79 F
31/03/96 : 171.78 F
 5/04/96 : 162.84 F
29/04/96 : 145.60 F
21/05/96 : 143.16 F
25/06/96 : 200.00 F
 2/07/96 : 141.00 F
 8/07/96 : 2104.97 F
13/07/96 : 186.00 F
15/07/96 : 116.00 F
15/07/96 : 94.10 F
20/07/96 : 122.80 F
26/07/96 : 154.55 F
```

КАК ЧИТАТЬ МАГНИТНЫЕ ПОЛОСЫ

У некоторых карт на микропроцессоре есть магнитные полосы (рис. 2.8), хотя бы частично воспроизводящие содержимое чипа, несмотря на то что таким образом существенно снижается безопасность.

Такая ситуация характерна для всех международных банковских карт. Она будет оставаться актуальной до тех пор, пока чипы не будут внедрены в мировом масштабе. Однако и «национальные» карты имеют к ней прямое отношение, поскольку французские банкоматы обязаны принимать у иностранных гостей карты только с магнитной полосой (без чипа).

На рис. 2.9 показано расположение трех «интернациональных» полос: ISO1, ISO2 и ISO3. Дополнительных полос T2 и T3, которые раньше находились на обратной стороне банковских карт VISA, сегодня уже нет.

Эти полосы могут быть раскодированы по очень простой схеме, представленной на рис. 2.10. Каждому изменению направления намагничивания соответствует вертикальная черта, прекрасно видная в лупу при нанесении на полосу специального «магнитного обнаружителя».

Плотность записи может составлять 75 или 210 бит/дюйм. Допустимо писать либо алфавитно-цифровым кодом по 7 бит (код ASCII), либо только цифровым – по 5 бит (с битом контроля четности).

Учитывая длину карты (приблизительно 85 мм), получают максимальное число знаков: 79 – для полосы ISO1 (также называемой IATA, так как ее часто используют авиакомпании), 40 знаков – для полосы ISO2 или ABA, широко применяемой банками, и 107 знаков – для полосы ISO3, использование которой в большей или меньшей степени произвольно. Это, конечно, немного по сравнению с современной чип-картой, но все же несопоставимо с «жалкими» 256 битами (а не знаками!) телефонной чип-карты.

Хотя вполне возможно и зрительно раскодировать (особенно при плотности записи 75 бит/дюйм) содержимое предварительно считанной магнитной полосы, гораздо практичнее воспользоваться специальном устройством для декодирования.

В книге "Cartes magnetiques et PC", вышедшей во Франции[1], расписано «от А до Я», как собрать и запустить коммерческие или бывшие в употреблении устройства для считывания магнитных карт. Кроме этого, объясняется, как сконструировать свое собственное устройство для чтения-декодирования на базе легко доступных элементов. Этого

---

[1] Русский перевод («Магнитные карты и ПК») готовится к выпуску в издательстве «ДМК».

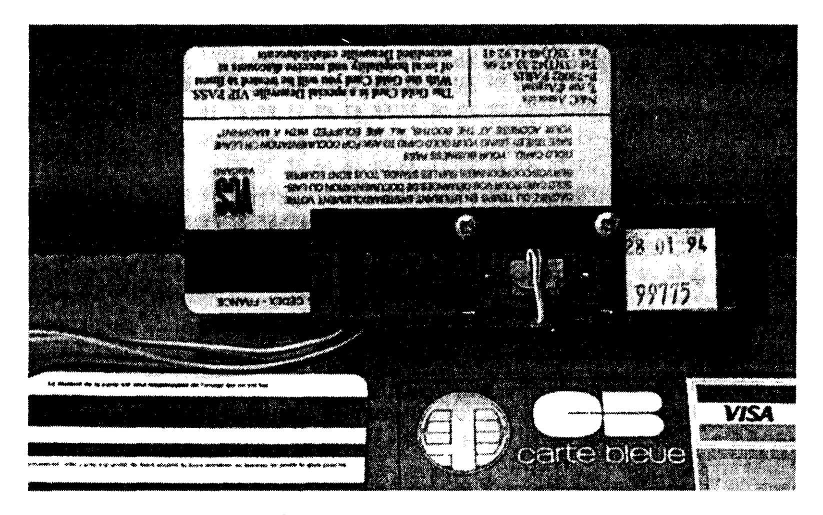


Рис. 2.8. Магнитные полосы

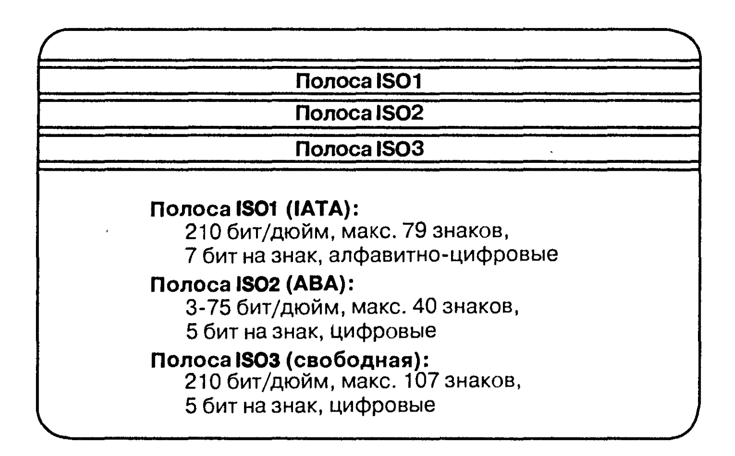


Рис. 2.9. Характеристики магнитных полос

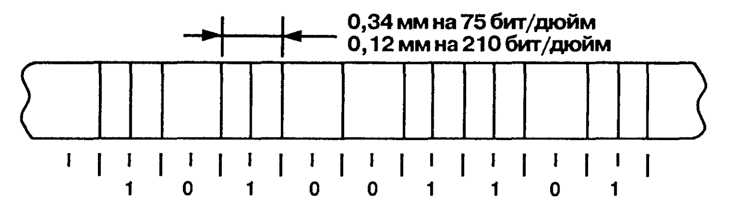


Рис. 2.10. Кодирование битов на магнитных полосах

хватит, чтобы продолжить исследования магнитных полос так же основательно, как исследования чипов.

ПЕРСПЕКТИВЫ

После банковских карт читатели, несомненно, захотят попрактиковаться и на других категориях карт на микропроцессоре (карты SIM для мобильных телефонов, электронные кошельки, медицинские страховые полисы и т.д.). Прежде чем пытаться разработать свой собственный набор инструментов для того или иного семейства карт, следует освоить совершенно незнакомый чип.

Первый этап состоит в том, чтобы получить ответ на сброс при помощи программ ATR.EXE и DECATR.EXE, которые можно найти в книге «Чип-карты. Устройство и применение в практических конструкциях». Если карта хорошо воспринимает протокол T=0, нужно определить ее как класс ISO. Здесь опять в дело вступают две программы, опубликованные в предыдущем издании: CLASSINV.EXE, если карта работает по обратному соглашению, или CLASSDIR.EXE, если ответ на сброс сообщает о том, что она работает по прямому соглашению.

На этой стадии лучше всего автоматически определить нужную систему команд, а не пробовать наудачу различные варианты. Это совсем несложно, так как в большинстве случаев достаточно запустить программы OPINV.EXE (при прямом соглашении) или OPDIR.EXE (при обратном), полученные при компиляции исходных текстов OPINV.BAS и OPDIR.BAS соответственно.

```
10 REM -- OPINV.©BAS --
20 KEY OFF :CLS
30 PRINT"Поиск системы команд для карт при обратном соглашении";
40 PRINT :PRINT
50 OPEN "COM1:9600,o,8,2" AS #1
60 PRINT"Вставить карту в картоприемник"
70 IF LOC(1)<>0 THEN 100
80 IF INKEY$=CHR$(27) THEN 570
90 GOTO 70
100 T$=TIME$ :BEEP
110 IF T$=TIME$ THEN 110
120 A$=""
130 WHILE NOT EOF(1)
140 A$=A$+INPUT$(1,#1)
150 WEND
160 IF LEFT$(A$,1) = CHR$(3) THEN 171
170 CLS :PRINT"Карта не опознана" :GOTO 570
171 PRINT :PRINT"Класс ISO карты ";
172 INPUT C$ :IF LEN(C$)<>2 THEN 172
```

```
173 PRINT :PRINT"Опробовать коды операций, начиная с (hex) ";
174 INPUT S$:S$="&H"+S$:S=VAL(S$)
180 FOR R=S TO 255
190 IF INKEY$=CHR$(27) THEN 570
200 R$=HEX$(R):D$=""
210 IF LEN(R$)=1 THEN R$="0"+R$
220 B$=C$+R$+"FFFFFF"
230 PRINT :PRINT R$;" ";
240 FOR F=1 TO 9 STEP 2
250 K$=MID$(B$,F,2)
260 N$="&h"+K$
270 N=VAL(N$)
280 GOSUB 580
290 PRINT#1,CHR$(M);
300 FOR T=0 TO 20 :NEXT T
310 NEXT F
320 T$=TIME$
330 IF T$=TIME$ THEN 330
340 IF LOC(1)<>0 THEN 370
350 IF INKEY$=CHR$(27) THEN 570
360 GOTO 340
370 A$=INPUT$(LOC(1),#1)
380 D$=""
390 FOR F=1 TO LEN(A$)
400 N$=MID$(A$,F,1)
410 N=ASC(N$)
420 GOSUB 580
430 M$=HEX$(M)
440 IF LEN(M$)=1 THEN M$="0"+M$
450 D$=D$+M$
460 NEXT F
470 IF LOC(1)<>0 THEN A$=INPUT$(LOC(1),#1) :GOTO 390
480 IF LEFT$(D$,2)="6E" THEN 170
482 PRINT D$;" ";
485 IF LEN(D$)>3 AND LEFT$(D$,2)<>"6D" THEN 510
490 NEXT R
500 GOTO 570
510 PRINT"Код операции опознан"; :BEEP
540 IF INKEY$=CHR$(27) THEN 570
550 GOTO 490
570 PRINT :PRINT :END
580 M=255
590 IF N>127 THEN N=N-128:M=M-1
600 IF N>63 THEN N=N-64:M=M-2
610 IF N>31 THEN N=N-32:M=M-4
620 IF N>15 THEN N=N-16:M=M-8
630 IF N>7 THEN N=N-8:M=M-16
640 IF N>3 THEN N=N-4:M=M-32
```

```
650 IF N>1 THEN N=N-2:M=M-64
660 IF N>0 THEN M=M-128
670 RETURN
680 REM (c)1997 Patrick GUEULLE
10 REM -- OPDIR.BAS --
20 KEY OFF :CLS
30 PRINT"Поиск системы команд для карт при прямом соглашении";
40 PRINT :PRINT
50 OPEN "COM1:9600,e,8,2" AS #1
60 PRINT"Вставить карту в картоприемник";
70 IF LOC(1)<>0 THEN 100
80 IF INKEY$=CHR$(27) THEN 550
90 GOTO 70
100 T$=TIME$ :BEEP
110 IF T$=TIME$ THEN 110
120 A$=""
130 WHILE NOT EOF(1)
140 A$=A$+INPUT$(1,#1)
150 WEND
160 IF LEFT$(A$,1) = CHR$(59) THEN 171
170 CLS :PRINT"Карта не опознана" :GOTO 550
171 PRINT :PRINT"Класс ISO карты ";
172 INPUT C$ :IF LEN(C$)<>2 THEN 172
173 PRINT :PRINT"Опробовать коды операций, начиная с (hex) ";
174 INPUT S$:S$="&H"+S$:S=VAL(S$)
180 FOR R=S TO 255
190 IF INKEY$=CHR$(27) THEN 550
200 R$=HEX$(R)
210 IF LEN(R$)=1 THEN R$="0"+R$
220 B$=C$+R$+"FFFFFF"
230 PRINT :PRINT R$;" ";
240 FOR F=1 TO 9 STEP 2
250 K$=MID$(B$,F,2)
260 N$="&h"+K$
270 M=VAL(N$)
280 PRINT#1,CHR$(M);
290 FOR T=0 TO 20 :NEXT T
300 NEXT F
310 T$=TIME$
320 IF T$=TIME$ THEN 320
330 IF LOC(1)<>0 THEN 360
340 IF INKEY$=CHR$(27) THEN 550
350 GOTO 330
360 A$=INPUT$(LOC(1),#1)
370 D$=""
380 FOR F=1 TO LEN(A$)
390 N$=MID$(A$,F,1)
400 M=ASC(N$)
```

```
410 M$=HEX$(M)
420 IF LEN(M$)=1 THEN M$="0"+M$
430 D$=D$+M$
440 NEXT F
450 IF LOC(1)<>0 THEN A$=INPUT$(LOC(1),#1) :GOTO 380
451 PRINT D$;" ";
455 IF LEFT$(D$,2)="6E" THEN 170
460 IF LEFT$(D$,2)<>"6D" THEN 490
470 NEXT R
480 GOTO 550
490 PRINT"Код операции опознан"; :BEEP
520 IF INKEY$=CHR$(27) THEN 550
530 GOTO 470
550 PRINT :PRINT :END
560 REM (c)1997 Patrick GUEULLE
```

Систематически пробуются все возможные коды операций, связанные с предварительно определенным классом ISO. Наконец, во избежание случайностей, проверка ведется по абсолютно невозможным параметрам «номер» и «длина» (соответственно FFFFh и FFh).

Логично приступить к исследованию начиная с кода операции 00h, но программа позволяет начать и с любого другого, в пределах от 00h до FFh. Таким образом допустимо «перепрыгнуть» коды, которые могут вызвать блокировку карты и/или перевести ее в режим молчания (распространенная реакция карт BULL CP8).

Полное исследование занимает менее 5 минут, в течение которых звуковой сигнал оповещает о каждом опознанном коде операции. Настоятельно рекомендуется записывать его.

В качестве иллюстрации ниже даны списки исполняемых кодов, которые автору удалось получить подобным способом.

Телефонная карта GSM:

04, 20, 24, 26, 28, 2C, 32, 44, 88, A2, A4, B0, B2, B8, BE, C0, D6, DA, DC, E0, E4, F2, FA

Электронный кошелек MONDEX:

20, 22, 24, 32, 34, 36, 38, 3C, 3E, 44, 46, 48, 4A, 4C, 4E, 50, 52, 54, 56, 70, 72, 74, 76, 80, 82, 84, 86, F6

Электронный кошелек PROTON:

10, 14, 1C, 20, 50, A0, A4, A8, B0, B2, C0, C4, D0, D4, D6

Карта COS компании Gemplus:

20, 22, 24, 26, 28, A2, B0, B2, B4, D0, D4, D8, E0, F0, F2

Далее надо определить функцию и синтаксис каждой найденной команды. Неоценимую услугу в этом могут оказать словари существующих кодов операций.

Сначала стоит попробовать некоторые команды с различными номерами (часто дает результаты 0000h), а затем – с различными длинами. Если карта отвечает протоколом, начинающимся с 6Bh, это означает, что данный номер не подходит. Когда же протокол начинается с 67h, неправильно подобрана длина.

Хотя в это трудно поверить, многие карты выдают на втором байте протокола истинную длину. Так, 67 20 означает, что ожидаемая длина данной системы составляет 20h, иначе говоря, 32 байта. В действительности длина контролируется таким образом только при правильном номере (или «адресе»).

Иногда создается впечатление, что карта очень логично начинает тестирование с класса ISO, затем проверяет исполняемый код, номер и, наконец, длину, хотя на самом деле сообщает она исключительно о первой встреченной неточности.

Как только ситуация прояснилась, надо попробовать прочитать блоки байтов по потенциально интересующим адресам. Если карта распознает команду чтения B0h, можно воспользоваться программой LECTIN.EXE (протокол по обратному соглашению) или LECTDIR.EXE (протокол по прямому соглашению). Ниже приведены их исходные тексты:

```
10 REM -- LECTINV.BAS --
20 KEY OFF :CLS
30 PRINT"LECTINV (c)1997 П. Гелль. Чтение карт с протоколом по обрат. согл. ISO"
40 PRINT :PRINT
50 OPEN "COM1:9600,o,8,2" AS #1
60 PRINT"Вставить карту в картоприемник "
70 IF LOC(1)<>0 THEN 100
80 IF INKEY$=CHR$(27) THEN END
90 GOTO 70
100 T=TIMER :BEEP
110 IF T>TIMER-1 THEN 110
120 A$=""
130 WHILE NOT EOF(1)
140 A$=A$+INPUT$(1,#1)
150 WEND
160 IF LEFT$(A$,1) = CHR$(3) THEN 180
170 CLS :PRINT"Карта по обратному соглашению не опознана" :END
180 CLS :PRINT"Число байтов для считывания (max. 250) ", :INPUT N
190 PRINT :PRINT"Начиная с адреса (hex) ", :INPUT A$
200 PRINT :PRINT"Класс ISO карты (hex) ", :INPUT C$
210 N$=HEX$(N)
220 IF LEN(A$)<4 THEN A$="0"+A$ :GOTO 220
```

```
230 IF LEN(N$)<2 THEN N$="0"+N$ :GOTO 230
240 IF LEN(C$)<2 THEN C$="0"+C$ :GOTO 240
250 B$=C$+"B0"+A$+N$
260 IF LEN(B$)<>10 THEN CLS :PRINT"Ошибка!" :BEEP :END
270 PRINT :PRINT"Команда, отправленная на карту: "
280 FOR F=1 TO 9 STEP 2
290 K$=MID$(B$,F,2)
300 PRINT K$;" ";:N$="&h"+K$
310 N=VAL(N$) :GOSUB 690
320 PRINT#1,CHR$(M);
325 FOR N=0 TO 20 :NEXT T
330 NEXT F
340 PRINT :PRINT :PRINT"Ответ карты:" :PRINT
350 T=TIMER
355 IF T>TIMER-1 THEN 355
360 IF LOC(1)<>0 THEN 390
370 IF INKEY$=CHR$(27) THEN END
3B0 GOTO 360
390 A$=INPUT$(LOC(1),#1)
400 D$=""
410 FOR F=1 TO LEN(A$)
420 N$=MID$(A$,F,1)
430 N=ASC(N$)
440 GOSUB 690
450 M$=HEX$(M)
460 IF LEN(M$)=1 THEN M$="0"+M$
470 PRINT M$+" ";:D$=D$+M$
480 NEXT F
490 IF LOC(1)<>0 THEN A$=INPUT$(LOC(1),#1) :GOTO 410
500 CLOSE#1
510 PRINT :PRINT :PRINT"Создание файла CARTE.CAR ";
520 OPEN "carte.car" FOR OUTPUT AS #1
530 K=0 :PRINT
540 FOR F=3 TO LEN(D$)-5 STEP 2
550 B$="&h"+MID$(D$,F,2)
560 B=VAL(B$)
570 IF(B AND 128) = 128 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
580 IF(B AND 64) = 64 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
590 IF(B AND 32) = 32 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
600 IF(B AND 16) = 16 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
610 IF(B AND B) = 8 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
620 IF(B AND 4) = 4 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
630 IF(B AND 2) = 2 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
640 IF(B AND 1) = 1 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
650 K=K+1
660 IF K>3 THEN PRINT#1,:K=0
670 NEXT F
```

```
680 END
690 M=255
700 IF N>127 THEN N=N-128:M=M-1
710 IF N>63 THEN N=N-64:M=M-2
720 IF N>31 THEN N=N-32:M=M-4
730 IF N>15 THEN N=N-16:M=M-8
740 IF N>7 THEN N=N-8:M=M-16
750 IF N>3 THEN N=N-4:M=M-32
760 IF N>1 THEN N=N-2:M=M-64
770 IF N>0 THEN M=M-128
780 RETURN
790 REM (c)1997 Patrick GUEULLE
```

```
10 REM -- LECTDIR.BAS --
20 KEY OFF :CLS
30 PRINT"LECTDIR (c)1997 П. Гелль. Чтение карт с протоколом по прям. согл. ISO "
40 PRINT :PRINT
50 OPEN "COM1:9600,e,8,2" AS #1
60 PRINT"Вставить карту в картоприемник"
70 IF LOC(1)<>0 THEN 100
80 IF INKEY$=CHR$(27) THEN END
90 GOTO 70
100 T=TIMER :BEEP
110 IF T>TIMER-1 THEN 110
120 A$=""
130 WHILE NOT EOF(1)
140 A$=A$+INPUT$(1,#1)
150 WEND
160 IF LEFT$(A$,1) = CHR$(59) THEN 180
170 CLS :PRINT"Карта по прямому соглашению не опознана" :END
180 CLS :PRINT"Число байтов для считывания (max. 250) ", :INPUT N
190 PRINT :PRINT"Начиная с адреса hexa " ,:INPUT A$
200 PRINT :PRINT"Класс ISO карты (hexa) " ,:INPUT C$
210 N$=HEX$(N)
220 IF LEN(A$)<4 THEN A$="0"+A$ :GOTO 220
230 IF LEN(N$)<2 THEN N$="0"+N$ :GOTO 230
240 IF LEN(C$)<2 THEN C$="0"+C$ :GOTO 240
250 B$=C$+"B0"+A$+N$
260 IF LEN(B$)<>10 THEN CLS :PRINT"Ошибка!" :BEEP :END
270 PRINT :PRINT"Команда, отправленная на карту: ",
280 FOR F=1 TO 9 STEP 2
290 K$=MID$(B$,F,2)
300 PRINT K$;" ";:N$="&h"+K$
310 M=VAL(N$)
320 PRINT#1,CHR$(M);
325 FOR T=0 TO 20 :NEXT T
330 NEXT F
```

```
340 PRINT :PRINT :PRINT"Ответ карты:" :PRINT
350 T=TIMER
355 IF T>TIMER-1 THEN 355
360 IF LOC(1)<>0 THEN 390
370 IF INKEY$=CHR$(27) THEN END
380 GOTO 360
390 A$=INPUT$(LOC(1),#1)
400 D$=""
410 FOR F=1 TO LEN(A$)
420 N$=MID$(A$,F,1)
430 M=ASC(N$)
440 M$=HEX$(M)
450 IF LEN(M$)=1 THEN M$="0"+M$
460 PRINT M$+" ";:D$=D$+M$
470 NEXT F
480 IF LOC(1)<>0 THEN A$=INPUT$(LOC(1),#1) :GOTO 410
490 CLOSE#1
500 PRINT :PRINT :PRINT"Создание файла CARTE.CAR ";
510 OPEN "carte.car" FOR OUTPUT AS #1
520 K=0 :PRINT
530 FOR F=3 TO LEN(D$)-5 STEP 2
540 B$="&h"+MID$(D$,F,2)
550 B=VAL(B$)
560 IF(B AND 128) = 128 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
570 IF(B AND 64) = 64 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
580 IF(B AND 32) = 32 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
590 IF(B AND 16) = 16 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
600 IF(B AND 8) = 8 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
610 IF(B AND 4) = 4 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
620 IF(B AND 2) = 2 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
630 IF(B AND 1) = 1 THEN PRINT#1,"1 "; ELSE PRINT#1,"0 ";
640 K=K+1
650 IF K>3 THEN PRINT#1,:K=0
660 NEXT F
670 END
680 REM (c)1997 Patrick GUEULLE
```

Можно изменить строку 250 и затем перекомпилировать программу так, чтобы воспользоваться любым другим исполняемым кодом, запускающим операцию считывания.

Технический прогресс приводит к тому, что время от времени встречаются карты, которые несовместимы с привычно используемым программным обеспечением. В таком случае возникает необходимость его модификации или разработки нового. Например, DIALINV.BAS – вариант INVERSE.BAS, преимущество которого заключается в совместимости с новыми поколениями карт (таких как электронный кошелек MONDEX и др.).

```
10 REM -- DIALINV.BAS --
20 KEY OFF :CLS
30 PRINT"DIALINV (c)1997 Патрик Гелль. Диалог с картами по обрат. соглашению"
40 PRINT :PRINT
50 OPEN "COM1:9600,o,8,2" AS #1
60 PRINT"Вставить карту в картоприемник "
70 IF LOC(1)<>0 THEN 100
80 IF INKEY$=CHR$(27) THEN END
90 GOTO 70
100 T=TIMER :BEEP
110 IF T>TIMER-1 THEN 110
120 A$=""
130 WHILE NOT EOF(1)
140 A$=A$+INPUT$(1,#1)
150 WEND
160 IF LEFT$(A$,1) = CHR$(3) THEN 180
170 CLS :PRINT"Карта по обратному соглашению не опознана" :END
180 PRINT :PRINT"Ввести команду для отправки на карту", :INPUT B$
185 IF B$="" THEN END
190 PRINT :PRINT"Команда, отправленная на карту: ",
200 FOR F=1 TO LEN(B$)-1 STEP 2
210 K$=MID$(B$,F,2)
220 PRINT K$;" ";:N$="&h"+K$
230 N=VAL(N$) :GOSUB 460
240 PRINT#1,CHR$(M);
250 FOR T=0 TO 20 :NEXT T
260 NEXT F
270 PRINT :PRINT :PRINT"Ответ карты:" :PRINT
280 T=TIMER
290 IF T>TIMER-1 THEN 290
300 IF LOC(1)<>0 THEN 330
310 IF INKEY$=CHR$(27) THEN END
320 GOTO 300
330 A$=INPUT$(LOC(1),#1)
340 D$=""
350 FOR F=1 TO LEN(A$)
360 N$=MID$(A$,F,1)
370 N=ASC(N$)
380 GOSUB 460
390 M$=HEX$(M)
400 IF LEN(M$)=1 THEN M$="0"+M$
410 PRINT M$+" ";:D$=D$+M$
420 NEXT F
430 IF LOC(1)<>0 THEN A$=INPUT$(LOC(1),#1) :GOTO 350
440 IF INKEY$=CHR$(27) THEN END
450 PRINT :GOTO 180
460 M=255
470 IF N>127 THEN N=N-128:M=M-1
```

```
480 IF N>63 THEN N=N-64:M=M-2
490 IF N>31 THEN N=N-32:M=M-4
500 IF N>15 THEN N=N-16:M=M-8
510 IF N>7 THEN N=N-8:M=M-16
520 IF N>3 THEN N=N-4:M=M-32
530 IF N>1 THEN N=N-2:M=M-64
540 IF N>0 THEN M=M-128
550 RETURN
560 REM (c)1997 Patrick GUEULLE
```

```
10 REM -- DIALDIR.BAS --
20 KEY OFF :CLS
30 PRINT"DIALDIR (c)1997 Патрик Гелль. Диалог с картами по прямому соглашению"
40 PRINT :PRINT
50 OPEN "COM1:9600,e,8,2" AS #1
60 PRINT"Вставить карту в картоприемник "
70 IF LOC(1)<>0 THEN 100
80 IF INKEY$=CHR$(27) THEN END
90 GOTO 70
100 T=TIMER :BEEP
110 IF T>TIMER-1 THEN 110
120 A$=""
130 WHILE NOT EOF(1)
140 A$=A$+INPUT$(1,#1)
150 WEND
160 IF LEFT$(A$,1) = CHR$(59) THEN 180
170 CLS :PRINT"Карта по прямому соглашению не опознана" :END
180 PRINT :PRINT"Ввести команду для отправки на карту" :INPUT B$
185 IF B$="" THEN END
190 PRINT :PRINT"Команда, отправленная на карту: "
200 FOR F=1 TO LEN(B$)-1 STEP 2
210 K$=MID$(B$,F,2)
220 PRINT K$;" "; :N$="&h"+K$
230 M=VAL(N$)
240 PRINT#1,CHR$(M);
250 FOR T=0 TO 20:NEXT T
260 NEXT F
270 PRINT :PRINT :PRINT"Ответ карты:" :PRINT
280 T=TIMER
290 IF T>TIMER-1 THEN 290
300 IF LOC(1)<>0 THEN 330
310 IF INKEY$=CHR$(27) THEN END
320 GOTO 300
330 A$=INPUT$(LOC(1),#1)
340 D$=""
350 FOR F=1 TO LEN(A$)
360 N$=MID$(A$,F,1)
370 M=ASC(N$)
```

```
390 M$=HEX$(M)
400 IF LEN(M$)=1 THEN M$="0"+M$
410 PRINT M$+" ";:D$=D$+M$
420 NEXT F
430 IF LOC(1)<>0 THEN A$=INPUT$(LOC(1),#1):GOTO 350
440 IF INKEY$=CHR$(27) THEN END
450 PRINT :GOTO 180
460 REM (c)1997 Patrick GUEULLE
```

Программа INVERSE.BAS передает команды на карту байт за байтом, по мере их набора на клавиатуре. Хотя такой способ и имеет право на существование, некоторые карты, разработанные недавно, задают очень короткий разрыв между двумя последующими байтами. Возможно, это объясняется необходимостью отслеживать реакцию карты на «специальные» запросы.

Программа DIALINV.BAS и ее вариант DIALDIR.BAS для карт по прямому соглашению решают указанную проблему, запоминая «набитые» байты и отправляя их на карту сразу после нажатия на клавишу ENTER. Следует напомнить, что из-за конфигурации последовательного порта исходные тексты на языке BASIC не должны использоваться напрямую из интерпретатора. Необходимо прибегнуть к откомпилированным версиям DIALINV.EXE и DIALDIR.EXE, которые можно найти на сайте www.dmk.ru, запуская их прямо из командной строки DOS или в одном из диалоговых окон Windows (только на достаточно быстродействующих ПК). Впрочем, исходные тексты могут оказаться полезными тем читателям, которые пожелают внести свои модификации в приведенные программы (автор будет благодарен, если его об этом проинформируют).

ГЛАВА СТРАНИЦА

3 МИНИ-СИСТЕМА РАЗРАБОТКИ

Можно считать, что работа асинхронной карты в большинстве случаев сводится к обмену байтов на скорости 9600 бод. При этом не учитывается большинство тонкостей ISO 7816-3, даже тех, на которые не может не обратить внимания разработчик коммерческих приложений.

Иногда возникает желание соединить карты на микропроцессоре, считывающие устройства и порты RS232 ПК для того, чтобы провести анализ протокола или даже (стоит только набраться храбрости!) сделать имитацию карты.

АДАПТЕР RS232 ДЛЯ АСИНХРОННЫХ КАРТ

Небольшая плата, схема которой представлена на рис. 3.2, использует только четыре из шести/восьми контактов устройства подключения чип-карты, хотя подавляющее большинство асинхронных карт использует шесть контактов, к одному из которых подведено напряжение программирования Vpp.

Может показаться странным отсутствие линии тактового генератора, но тому есть объяснение: ПК настраивает последовательный порт на 9600 бод на основе своего собственного внутреннего тактового

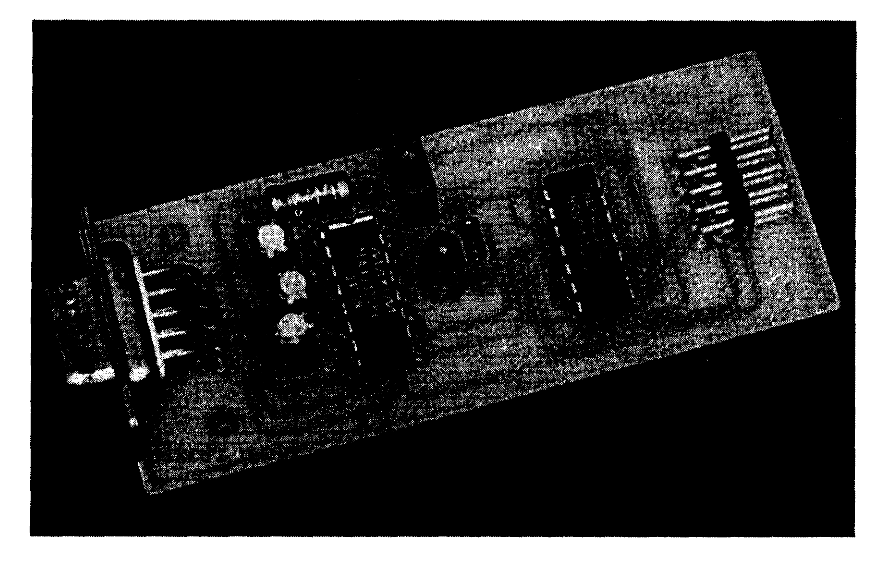


Рис. 3.1. Адаптер RS232

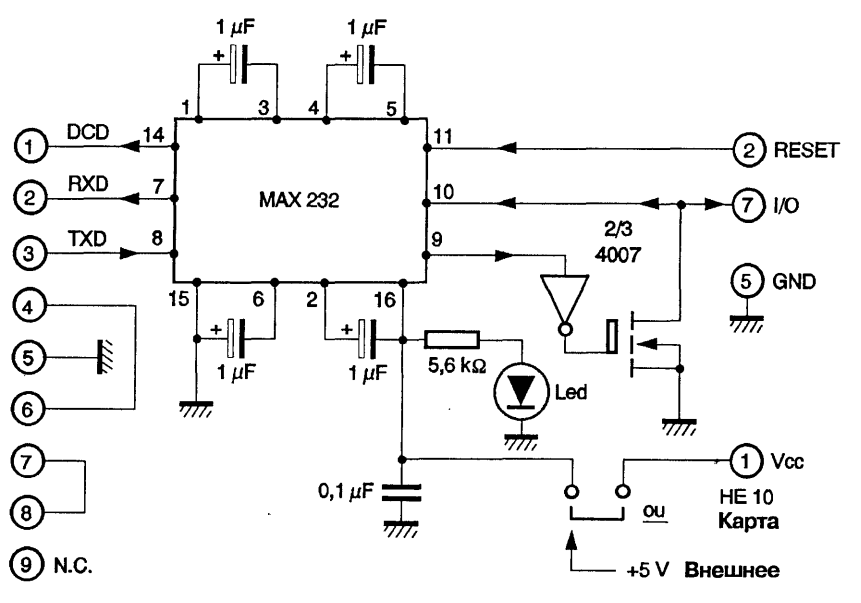


Рис. 3.2. Схема адаптера RS232

генератора, что позволяет ему полностью игнорировать сигнал CLK, поступающий с устройства чтения-записи. Это не совсем верно с точки зрения нормы ISO, которая предусматривает возможность изменения тактовой частоты и скорости передачи. Однако опыт показывает, что отсутствие генератора вполне приемлемо в большинстве случаев, представляющих потенциальный интерес. Основной частью блока является интегральная схема MAX232, которая получает питание +5 В от устройства чтения-записи. Она сама генерирует напряжения +12 В и –12 В, необходимые для работы с портом RS232. Для устройств чтения-записи, которые не могут обеспечить ток, достаточный для питания схемы (светодиод не загорается и/или устройство подает сигнал о коротком замыкании в карте), предусмотрен *джампер*. В вышеописанных ситуациях его снимают и подключают питание 5 В, которое подается на общий провод (отрицательный полюс) и на штырек разъема джампера, соединенный с выводом 16 интегральной микросхемы MAX 232 (положительный полюс). Джампер можно оставить «насаженным» на оставшийся свободным штырек, который соединен с контактом ISO1 чип-карты.

Данный метод исключает приложение внешнего напряжения питания, когда джампер находится на месте, и, следовательно, обеспечивает защиту устройства чтения-записи от подобных ошибочных действий.

Интегральную схему MAX 232 дополняет КМОП микросхема CD4007, которая нужна для мультиплексирования общей (двунаправленной) линии ввода/вывода карты, так как по линиям TXD и RXD адаптера RS232 данные циркулируют раздельно. Понятно, что эта конструкция помимо устройства для анализа протокола может, по крайней мере частично, имитировать карту на микропроцессоре!

Предложенная схема очень точно воспроизводит структуру с *открытым истоком* обычных чип-карт.

Преобразованный в уровни RS232 сигнал обнуления карты (RESET) поступает на ПК по линии DCD (Carrier detect) последовательного порта. Это позволяет программному обеспечению обнаружить запросы на обнуление, сформулированные устройством чтения-записи карт, и ответить на них.

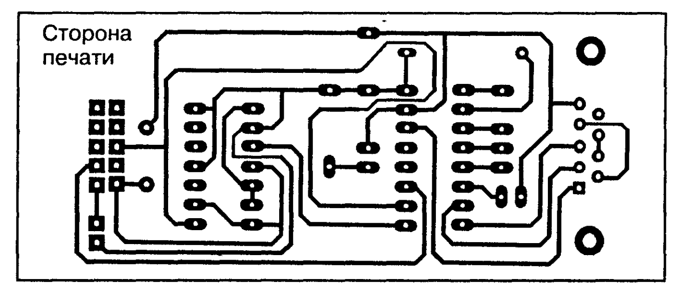


Рис. 3.3. Печатная плата адаптера RS232

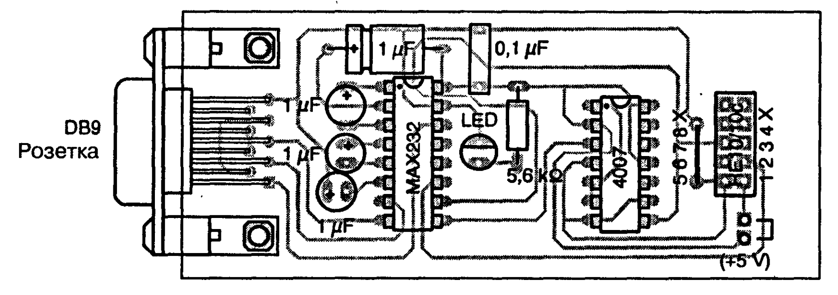


Рис. 3.4. Схема размещения элементов на плате адаптера RS232

Собранная на небольшой печатной плате схема (рис. 3.3) имеет два разъема, устанавливаемых в соответствии с рис. 3.4: разъем DB9 типа «розетка» подключается напрямую или при помощи адаптера DB9 – DB25 к порту COM1 ПК.

Внимание! Ни в коем случае нельзя использовать разъем типа «вилка» и кабель с двумя «розетками». Лучше применить кабель «вилка-розетка», также называемый *удлинителем*, и разъем HE10 типа «розетка», совместимый с блоком картоприемника, «фальшивыми картами» (в виде печатных плат из фольгированного материала) и, конечно, с устройствами чтения-записи, то есть со всеми элементами набора рабочих инструментов для чип-карт.

Очень важно обеспечить соответствие соединений между всеми элементами. Это условие будет выполняться автоматически, если установить на кабель с десятью проводниками несколько разъемов HE10 типа «розетка», а на платах использовать разрезные двухрядные колодки с соответствующим числом контактов. При неправильной ориентации разъема выступ на корпусах разъемов HE10 зацепится за печатную плату.

МАЛОГАБАРИТНЫЙ АНАЛИЗАТОР ПРОТОКОЛА

Интересно узнать, о чем могут «разговаривать» чип-карты с устройствами чтения-записи. Небольшой схемы и нескольких строк на языке BASIC достаточно, чтобы «раскрутить» банковскую, телефонную

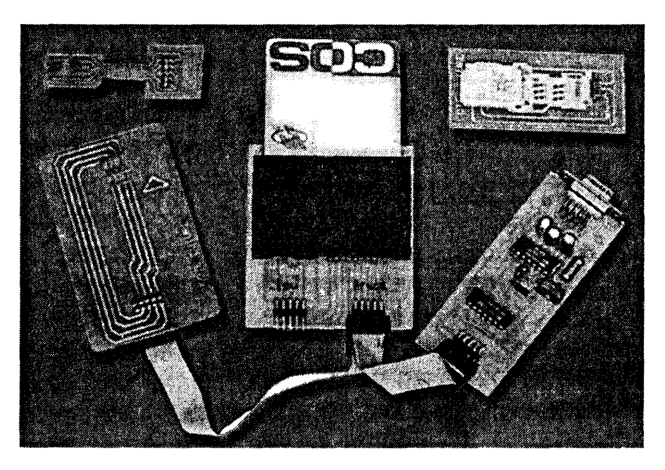


Рис. 3.5. Анализатор протокола и его модули

(PASTEL или GSM) карту или карты для платного телевидения. Можно даже представить диалог, перехваченный подобным образом.

Автор взял на себя смелость писать на эту тему, будучи уверенным в том, что не нанесет ущерба безопасности приложений на картах, защищенных соответствующим образом. Одна из задач данной книги заключается в том, чтобы дать читателям возможность составить собственное мнение по данному вопросу.

Хорошая платформа для экспериментов с асинхронной картой включает кроме указанного блока устройство чтения-записи для чип-карт (рис. 2.4), подключенное к ПК. Однако не стоит сразу задумываться об изучении домашнего оборудования наподобие Minitel с устройством чтения-записи, мобильного телефона GSM или декодера платного телевидения.

Как правило, для удобства работы требуются два компьютера, причем один из них (предпочтительно тот, который связан с анализатором) вполне может быть XT 8086 или ноутбуком.

С технической точки зрения, приспособление чрезвычайно просто: достаточно установить третий разъем HE10 с десятью контактами приблизительно в середине кабеля, соединяющего блок картоприемника (рис. 2.6) с «фальшивой картой» в виде печатной платы толщиной 0,8 мм. Топология подобной карты приведена на рис. 3.6 (чип в положении AFNOR).

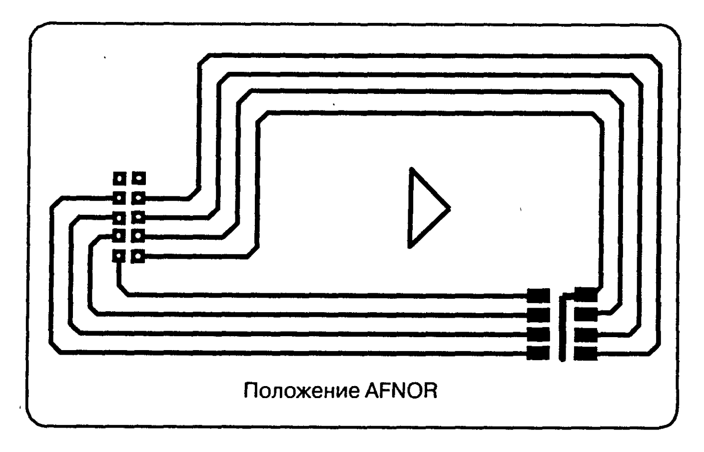


Рис. 3.6. Топология печатной платы «фальшивой карты» AFNOR

Очевидно, что третий разъем подключается к только что описанной конструкции.

Настоящая чип-карта (банковская или PASTEL, желательно просроченная) вставляется в блок картоприемника, соединенный с кабелем, затем в изучаемое устройство чтения-записи помещается «фальшивая карта» – теперь все подключено параллельно.

Для того чтобы показать в шестнадцатеричной системе все байты, циркулирующие между устройством чтения-записи и картой, надо запустить программу до выдачи устройством чтения-записи команды на сброс для карты:

```
10 REM -- ESPINV.BAS --
20 KEY OFF :CLS
30 PRINT"ESPINV (c) 1995,97 Патрик Гелль. Слежение за картами обрат. соглашения"
40 OPEN "COM1:9600,e,8,2" AS #1
50 E = INP(&H3FE) AND 128
60 IF E<>128 THEN 50
70 GOSUB 90
80 END
90 IF LOC(1)<>0 THEN GOSUB 110
100 GOTO 90
110 IF LOC(1)=0 THEN RETURN
120 C$=INPUT$(LOC(1),#1): PRINT
130 FOR K=1 TO LEN(C$)
140 N=ASC(MID$(C$,K,1))
150 M=255
160 IF N>127 THEN N=N-128:M=M-1
170 IF N>63 THEN N=N-64:M=M-2
180 IF N>31 THEN N=N-32:M=M-4
190 IF N>15 THEN N=N-16:M=M-8
200 IF N>7 THEN N=N-8:M=M-16
210 IF N>3 THEN N=N-4:M=M-32
220 IF N>1 THEN N=N-2:M=M-64
230 IF N>0 THEN M=M-128
240 D$=HEX$(M)+" "
250 IF LEN(D$)<3 THEN D$="0"+D$
260 PRINT D$
270 NEXT K
280 IF LOC(1)<>0 THEN C$=INPUT$(LOC(1),#1) :GOTO 130
290 PRINT
300 RETURN
310 REM (c)1995,97 Patrick GUEULLE
```

```
10 REM -- ESPDIR.BAS --
20 KEY OFF :CLS
30 PRINT"ESPDIR (c) 1995,97 Патрик Гелль. Слежение за картами прям. согл."
40 OPEN "COM1:9600,e,8,2" AS #1
50 E = INP(&H3FE) AND 128
60 IF E<>128 THEN 50
70 GOSUB 90
80 END
90 IF LOC(1)<>0 THEN GOSUB 110
100 GOTO 90
110 IF LOC(1)=0 THEN RETURN
120 C$=INPUT$(LOC(1),#1) :PRINT
130 FOR K=1 TO LEN(C$)
140 N=ASC(MID$(C$,K,1))
150 D$=HEX$(N)+" "
160 IF LEN(D$)<3 THEN D$="0"+D$
170 PRINT D$
180 NEXT K
190 IF LOC(1)<>0 THEN C$=INPUT$(LOC(1),#1) :GOTO 130
200 PRINT
210 RETURN
220 REM (c)1995,97 Patrick GUEULLE
```

Были разработаны две различные версии: ESPINV.BAS для карт по обратному соглашению и ESPDIR.BAS по прямому. На практике используют откомпилированные версии ESPINV.EXE и ESPDIR.EXE, размещенные на сайте www.dmk.ru.

В файле ESPINV.BAS содержится стандартная программа транскодирования байтов, в которой биты с большим весом в момент приема меняются с битами, имеющими меньший вес, в то же время инвертируя их. Этот процесс называется *преобразованием по обратному соглашению ISO*. Чип-карты такого типа передают информацию старшими разрядами вперед и в отрицательной логике (логическая единица представлена низким уровнем).

Как только поток данных прерывается (даже на короткое время), программа выполняет «перевод строки – возврат каретки». Тем самым она позволяет определять происхождение той или иной группы байтов, приходящих от устройства чтения-записи и от карты. На этой стадии экспериментирования удобно использовать с устройством чтения-записи программу, задействующую максимум функций банковских карт; чаще всего рекомендуется INVERSE.EXE.

Программа CB2PIN.BAS позволяет «добраться» до конфиденциального кода, используя для этого восемь байт, перехваченных в момент их передачи. Естественно, реализовать такую возможность можно лишь при достаточном основании.

```
10 REM -- CB2PIN.BAS --
20 KEY OFF :CLS
30 INPUT "Код hex, представленный карте: ",P$:K$=""
40 FOR F=1 TO LEN(P$)
50 M$="&h"+MID$(P$,F,1)
60 M=VAL(M$)
70 D=0
80 IF M>7 THEN D=1:M=M-8
90 GOSUB 300
100 IF M>3 THEN D=1:M=M-4
110 GOSUB 300
120 IF M>1 THEN D=1:M=M-2
130 GOSUB 300
140 D=M :GOSUB 300
150 NEXT F
160 K$=MID$(K$,3,LEN(K$)-16)
170 PRINT :PRINT"PIN = ";
180 FOR F=1 TO LEN(K$) STEP 4
190 D$=MID$(K$,F,4)
200 A=0
210 FOR G=1 TO 4
220 B$=MID$(D$,5-G,1)
230 IF B$="1" THEN A=A+2^(G-1)
240 NEXT G
250 A$=HEX$(A)
260 PRINT A$;
270 NEXT F
280 PRINT :PRINT :PRINT
290 END
300 IF D=0 THEN K$=K$+"0"
310 IF D=1 THEN K$=K$+"1"
320 D=0:RETURN
330 REM (c)1994 Patrick GUEULLE
```

МАЛОГАБАРИТНЫЙ ИМИТАТОР КАРТ

Среди многообразных инструментов, которыми располагает разработчик приложений чип-карт, особое место занимает *эмулятор* карт. Автономный или подключенный к ПК, этот инструмент, снабженный «фальшивой картой» из печатной платы, способен выдать себя

за любую *асинхронную* чип-карту – иначе говоря, карту на микропроцессоре.

В данной книге рассматривается очень простой эмулятор, который можно с большим основанием назвать *имитатором.* Это устройство также управляется ПК. В паре с анализатором протокола имитатор работает совместно с адаптером RS232.

С технической точки зрения, имитатор чип-карт может быть необыкновенно сложным или удивительно простым, но так или иначе обязательно нужны специальные программы для каждой конкретной карты, которую надо имитировать. Специалисты называют этот процесс *программированием маски.*

Можно сделать вывод, что достаточно собрать небольшую электронную схему и написать несколько строк кода для того, чтобы создать «истинно фальшивую» банковскую карту или карту для платного телевидения. Однако лучше этого не делать. На самом деле все карты на микропроцессоре, предназначенные для тонких приложений, должны, как правило, иметь криптографические функции, основанные на секретных ключах. Вне карты их нельзя прочитать ни при каких условиях (см. главу 1).

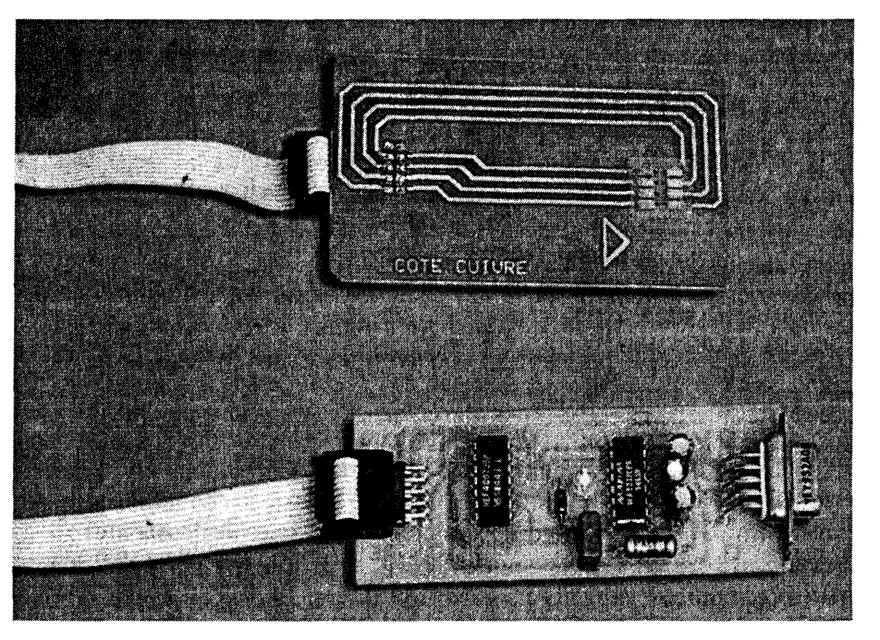


Рис. 3.7. Имитатор карты, готовой для подключения к ПК

Общепризнана невозможность проникновения в тайну защищенной части обмена данных между картой и устройством чтения-записи, если не была допущена грубая ошибка дистрибьютором карты (за что, разумеется, он несет полную ответственность).

Предлагаемый эмулятор, упрощенный до предела, тем не менее позволяет воспроизводить с помощью ПК не слишком секретные фрагменты диалога между картой и ее считывающим устройством, перехват которых был предварительно осуществлен. Остальное на совести читателя...

Для запуска имитатора к модулю адаптера RS232 надо подключить «фальшивую карту» (в виде печатной платы 0,8 мм) при помощи кабеля ad hoc.

Теперь следует подключить комплект к последовательному порту COM1 компьютера, причем использовать какой-либо картоприемник не требуется.

Манипуляции можно начать с выполнения короткой программы CARS232.BAS. Ее роль сводится к имитации правдоподобного ответа на сброс. Затем она генерирует в шестнадцатеричных кодах информацию, которую устройство чтения-записи впоследствии передает на карту. Таким образом, устройство «удостоверится», что ему представлена асинхронная карта, и первый этап исследований будет пройден.

```
10 REM -- CARS232.BAS --
20 KEY OFF
30 A$=CHR$(&H3)+CHR$(&HC9)+CHR$(255)+CHR$(255)
40 B$="AAAAAAAAAAAA"
50 OPEN "COM1:9600,n,8,2" AS #1
60 E = INP(&H3FE) AND 128
70 IF E<>128 THEN 60
80 PRINT #1,A$+B$
90 GOSUB 110
100 END
110 IF LOC(1)<>0 THEN :GOSUB 130
120 GOTO 110
130 IF LOC(1)=0 THEN :RETURN
140 ON ERROR :GOTO 340
150 C$=INPUT$(LOC(1),#1) :PRINT
160 FOR K=1 TO LEN(C$)
170 N=ASC(MID$(C$,K,1))
180 M=255
190 IF N>127 THEN N=N-128:M=M-1
200 IF N>63 THEN N=N-64:M=M-2
210 IF N>31 THEN N=N-32:M=M-4
```

```
220 IF N>15 THEN N=N-16:M=M-8
230 IF N>7 THEN N=N-8:M=M-16
240 IF N>3 THEN N=N-4:M=M-32
250 IF N>1 THEN N=N-2:M=M-64
260 IF N>0 THEN M=M-128
270 D$=HEX$(M)+" "
280 IF LEN(D$)<3 THEN D$="0"+D$
290 PRINT D$
300 NEXT K
310 IF LOC(1)<>0 THEN C$=INPUT$(LOC(1),#1) :GOTO 160
320 PRINT
330 RETURN
340 RESUME
350 REM (c)1995 Patrick GUEULLE
```

Относительно данной программы необходимо сделать несколько замечаний. Многие из ее элементов послужат базой для других программ, которые, возможно, потребуется написать впоследствии на BASIC или другом, более совершенном языке. В основном речь идет о замечаниях, актуальных для ESPINV.BAS: открытие последовательного порта ПК в режиме N (то есть без бита равенства), но с двумя стоп-битами.

После изменения настройки на 9600, e, 8, 2 желательно провести компиляцию на TURBO-BASIC.

Хитрость, используемая для того, чтобы версия GWBASIC могла работать на прием (ON ERROR, RESUME), неприменима в обратном направлении. Это обусловлено пунктуальностью устройств чтения-записи карт, соответствующих норме ISO 7816. При обнаружении ошибки четности они отправляют на карту продолжительный сигнал низкого уровня, затем ожидают повторения знака, который рассматривается как ошибочный. В данном случае возникает блокировка. Как следствие, при составлении ответа на сброс необходимо ограничиться использованием знаков, содержащих парное число битов 1. Это справедливо, например, для буквы А.

Стандартная операция в строках 60 и 70 – это реализация цикла ожидания команды на сброс. Возможно также ее повторное использование в программах для того, чтобы «фальшивая карта» правильно реагировала на запросы сброса, получаемые не только в начале, но и в процессе сессии.

Как и в программе ESPINV.BAS, обращает на себя внимание операция преобразования в обратное соглашение ISO, необходимая только

при приеме. В момент передачи преобразование ведется непосредственно при составлении ответа на сброс (табл. 3.1). После получения предварительного ответа на сброс считывающее устройство выдает карте команду: ответить.

Таблица 3.1. Преобразования в обратное соглашение

0	0000	1111	F
1	0001	0111	7
2	0010	1011	B
3	0011	0011	3
4	0100	1101	D
5	0101	0101	5
6	0110	1001	9
7	0111	0001	1
8	1000	1110	E
9	1001	0110	6
A	1010	1010	A
B	1011	0010	2
C	1100	1100	C
D	1101	0100	4
E	1110	1000	8
F	1111	0000	0
Прямое соглашение		Обратное соглашение	

Этим ограничены возможности программы, которая распечатывает приказ, исходящий от устройства чтения-записи, но не отвечает на него. Однако благодаря *анализатору протокола* известно, что должна ответить карта в ряде ситуаций; именно такие ответы и представляют интерес.

Программа SIMU.BAS позволяет воспроизводить на GWBASIC несколько основных функций карты типа COS: сброс, считывание двух байт DDDDh по адресу AAAAh и представление конфиденциального кода CCCCh по адресу BBBBh, если класс ISO равен 00h.

```
10 REM -- SIMU.BAS --
20 KEY OFF
30 A$=CHR$(&H3)+CHR$(&HC9)+CHR$(255)+CHR$(255)
40 B$="AAAAAAAAAAAA"
50 J$=CHR$(&HF6)+CHR$(&HFF)
60 OPEN "COM1:9600,N,8,2" AS #1
70 E=INP(&H3FE) AND 128
80 IF E<>128 THEN 70
90 PRINT#1,A$+B$
100 GOSUB 160
```

```
110 IF M$="00 B0 AA AA 02 " THEN GOSUB 390
120 IF M$="00 20 BB BB 02 " THEN GOSUB 400
130 IF M$="21 CC CC " THEN GOSUB 410
140 GOTO 100
150 END
160 IF LOC(1)=0 THEN 160
170 ON ERROR GOTO 370
180 M$=""
190 C$=INPUT$(LOC(1),#1) :PRINT
200 FOR K=1 TO LEN(C$)
210 N=ASC(MID$(C$,K,1))
220 M=255
230 IF N>127 THEN N=N-128:M=M-1
240 IF N>63 THEN N=N-64:M=M-2
250 IF N>31 THEN N=N-32:M=M-4
260 IF N>15 THEN N=N-16:M=M-8
270 IF N>7 THEN N=N-8:M=M-16
280 IF N>3 THEN N=N-4:M=M-32
290 IF N>1 THEN N=N-2:M=M-64
300 IF N>0 THEN M=M-128
310 D$=HEX$(M)+" "
320 IF LEN(D$)<3 THEN D$="0"+D$
330 M$=M$+D$
340 NEXT K
350 PRINT M$
360 RETURN
370 RESUME
380 REM (c)1995 Patrick GUEULLE
390 PRINT#1,CHR$(&H72)+"DD"+J$ :RETURN
400 PRINT#1,CHR$(&H7B) :RETURN
410 PRINT#1,J$ :RETURN
```

Необходимо отметить, что первые байты имитированного ответа на сброс информируют устройство чтения-записи о том, что карта функционирует только при напряжении 5 В (внешнее напряжение программирования Vpp отсутствует) и по обратному соглашению.

Обычно эту программу прокручивают на первом ПК, снабженном имитатором, в то время как на втором установлено устройство чтения-записи, собранное в соответствии с рис. 2.4. Естественно, SIMU.BAS должна запускаться прежде, чем программа, управляющая устройством чтения-записи, или, по крайней мере, до отправления ему любой команды о подключении напряжения к карте.

Важно, чтобы компьютеры правильно соотносились по быстродействию. Если, например, эмулятор работает на старом XT на 4,77 МГц,

желательно, чтобы тактовая частота ПК, работающего с устройством чтения-записи, составляла не более 8 МГц (XT на 8 МГц или AT386SX25 без «турбо»). Действительно, при обменах данными между картой и считывающим устройством предусмотрены многочисленные задержки. Их превышение создает большой риск возникновения блокировки (протокол *карта молчит* или *connector молчит*).

В любом случае программное обеспечение, упрощенное до такой степени, претендует только на очень приблизительную и, главное, примитивную имитацию чрезвычайно сложного процесса функционирования карты на микропроцессоре. Оно может лишь наметить путь, по которому следует продвигаться, чтобы написать собственные маски, написанные на языке по выбору исследователя. При желании создать автономную «фальшивую карту» на базе распространенного микроконтроллера начинают с написания масок на языке ассемблера.

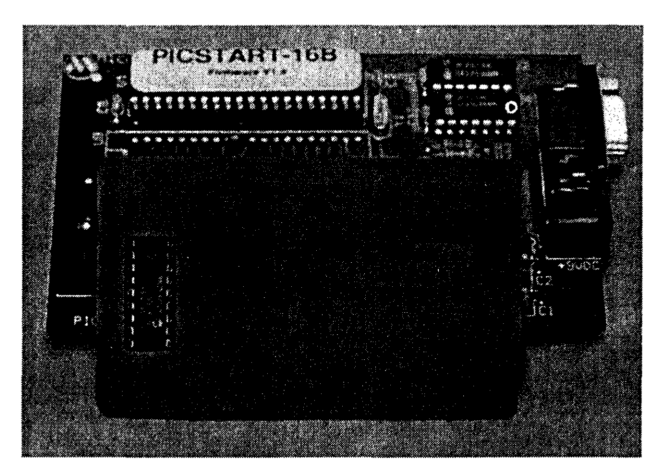


Рис. 3.8. Сторона элементов карты на PIC16C84

ЭКСПЕРИМЕНТАЛЬНАЯ ЧИП-КАРТА НА PIC16CXX

Асинхронная чип-карта (карта на микропроцессоре) – это не что иное, как специальный микроконтроллер в необычном корпусе, который представляет собой пластиковую карту, снабженную контактами в соответствии со стандартом ISO 7816.

Как это ни парадоксально с первого взгляда, достаточно подключить микроконтроллер в корпусе DIP или CMD к печатной плате толщиной 0,8 мм, чтобы получить «фальшивую карту», позволяющую проводить интереснейшие манипуляции, при этом не нарушая закон.

Рис. 3.9. Сторона печати карты на PIC16C84

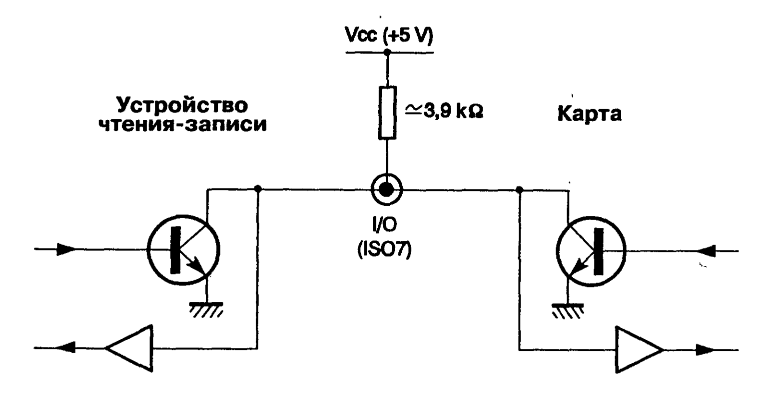


Рис. 3.10. Организация ввода/вывода по схеме ИЛИ

Контроллеры типа PIC16C71 и PIC16C84 компании MICROCHIP превосходно подходят для такого рода упражнений, поскольку, как будет показано ниже, для имитации базовых функциональных возможностей асинхронных карт требуется малое количество аппаратных и программных ресурсов.

Часто используемый термин *асинхронная карта* предполагает, что карта довольствуется получением и отправлением по очереди байтов, переданных в асинхронном последовательном режиме (half-duplex). Управление обменом подобными сообщениями доверено тактовому сигналу, подаваемому на карту от устройства чтения-записи, в которое ее вставляют таким образом, что при тактовой частоте приблизительно 3,58 МГц скорость обмена информацией составит 9600 бит/с.

Во избежание конфликтов аппаратного обеспечения нужна одна универсальная линия (контакт «ввод/вывод»), служащая для передачи байтов, получаемых от карты или предназначенных для нее, а также схема ИЛИ. На рис. 3.11 показано, как у устройства чтения-записи и карты расположены выходные каскады с открытым коллектором (или открытым стоком). Сопротивление нагрузки на Vpp должно быть предусмотрено, по крайней мере, со стороны устройства чтения-записи.

Когда линия передачи находится в состоянии лог. 1 (состоянии покоя), устройство чтения-записи или карта могут выдать низкий уровень (старт-бит), таким образом сигнализируя о начале передачи.

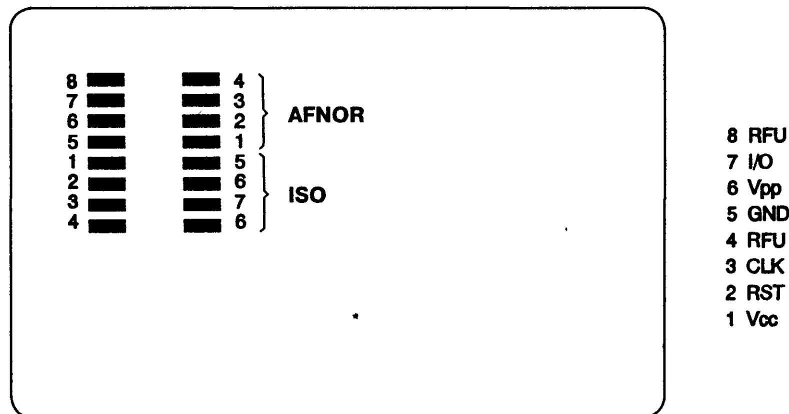


Рис. 3.11. Соответствие контактов по стандарту ISO 7816

Второе действующее лицо по правилам хорошего тона должно воздерживаться от одновременной передачи, хотя подобного рода конфликт не будет иметь разрушительных последствий ни для одной, ни для другой стороны.

Параметры связи определяются стандартом ISO 7816: старт-бит, восемь бит данных, бит контроля четности, стоп-бит, служащий также для обозначения ошибки передачи (низкий уровень, превышающий по длительности стоп-бит, представляет собой запрос на повторную передачу байта, полученного с ошибкой).

В любом случае допустимы два варианта: обратное соглашение ISO (старшими разрядами вперед, отрицательная логика) или, реже, прямое соглашение ISO (младшими разрядами вперед, положительная логика). Именно карта определяет один из вариантов для устройства чтения-записи в начале передачи ее ответа на сброс.

Группа байтов, спонтанно переданная картой при подключении к напряжению и/или в момент подачи сигнала на контакт сброса, содержит многочисленные сведения о параметрах карты, что позволяет устройству чтения-записи самонастроиться.

Стандартом ISO 7816 определено конкретное значение каждого байта ответа на сброс. Исключение, однако, составляют последние символы (так называемая *история*), использованием которых распоряжается разработчик приложения.

Для соединения карты с устройством чтения-записи достаточно пяти контактов:

- «земля» (GND) – ISO5;
- питание +5 В (Vcc) – ISO1;
- тактовый генератор (CLK) – ISO3;
- линия данных (ввод/вывод) – ISO7;
- вход reset (RST) – ISO2.

Для некоторых старых типов карт (с памятью ППЗУ) может понадобиться шестой контакт, обеспечивающий подачу напряжения программирования (Vpp) – ISO6. Два других контакта (ISO4 и ISO8) остаются в резерве для использования в будущем (RFU). Нумерация этих контактов в обязательном порядке определяется стандартом ISO 7816, но существует возможность выбора из двух вариантов (ISO и AFNOR). Обратившись к рис. 3.12, вы получите точное представление о том, что обеспечивает соответствие между контактами карты и блоком подключения устройства чтения-записи.

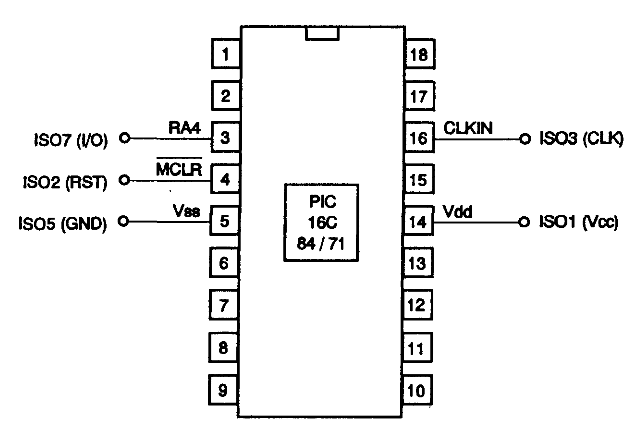


Рис. 3.12. Простейшая схема чип-карты на PIC16CXX

Что касается программного обеспечения, вполне возможно разработать *маску* асинхронной карты с использованием системы команд любого микроконтроллера. Автор предпочел быстродействующую архитектуру RISC микроконтроллеров семейства PIC16CXX. Конечно, 1024 слова ППЗУ, 36 байт ОЗУ и 64 байта ЭСППЗУ модели PIC16C84 кажутся несколько урезанными по сравнению с 3 Кбайт ПЗУ, 128 байт ОЗУ и 1 Кбайт ЭСППЗУ микросхемы 68HC055SC24, которая используется, например, в последних поколениях банковских карт (маска M4 B0').

Необходимо отметить, что код, написанный для микроконтроллеров PIC, значительно компактнее того, который предназначен для стандартных процессоров (так называемых CISC). Ничто не запрещает сочетать внешнее стандартное ЭСППЗУ с любым чипом, например типа 16C71 или семейства 16C5X, или даже заставить работать оба чипа параллельно на одной карте.

В любом случае множество не очень «тонких» приложений довольствуется лишь частью возможностей карт на стандартном микроконтроллере. Но в намерения автора книги не входил рассказ о том, как создать «фальшивую» банковскую карту или карту для платного телевидения.

А теперь давайте обратимся к операционной мини-системе, уже разработанной для микроконтроллеров PIC16C71 и PIC16C84 (примерно сотня слов, записываемых в ППЗУ) и предполагающей использование схемы, которая показана на рис. 3.13. Можно ли представить что-либо более простое?

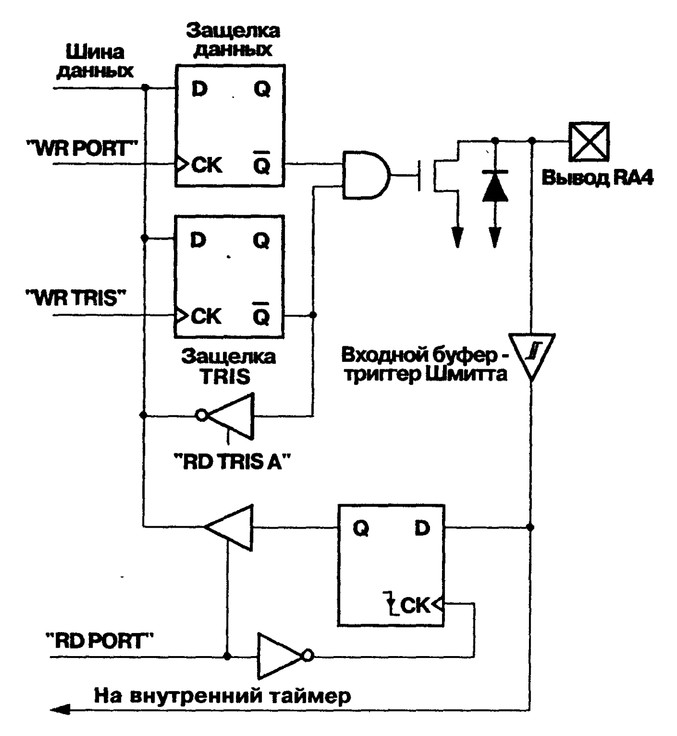


Рис. 3.13. Эквивалентная схема линии порта RA4 PIC16C84/71

Выводы «земли» и питания подключаются к соответствующим контактам ISO без развязывающего конденсатора (в «настоящих» чип-картах он отсутствует). Именно вход CLKIN напрямую принимает тактовый сигнал, поступающий с устройства чтения-записи: нет необходимости ни в кварце, ни даже в RC-цепи.

Сигнал *reset* поступает на вывод //MCLR (master clear) чипа, в то время как вывод RA4 зарезервирован для последовательного ввода-вывода.

Благодаря особенностям электрических характеристик, указанных на рис. 3.14, имеется возможность создать мощную схему ИЛИ для связи с вводом-выводом устройства чтения-записи.

Теперь остается создать «фальшивую карту» для проведения экспериментов. Были изучены два рисунка печатной платы: первый (рис. 3.15) имеет контакты в соответствии со стандартом ISO – европейским, а следовательно, наиболее перспективным; второй, изображенный

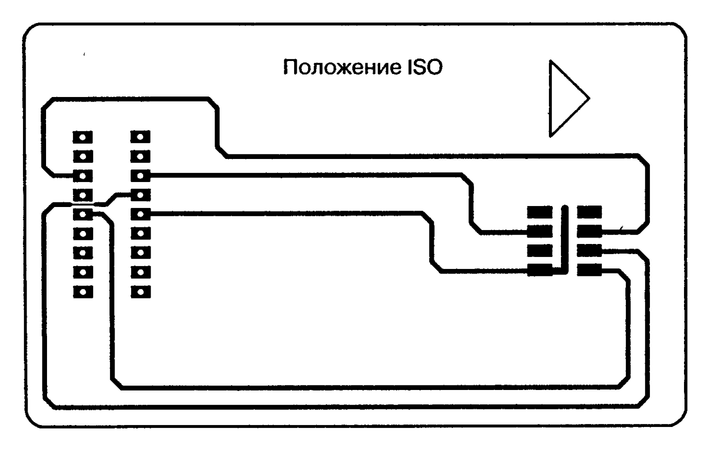


Рис. 3.14. Топология «фальшивой карты» ISO

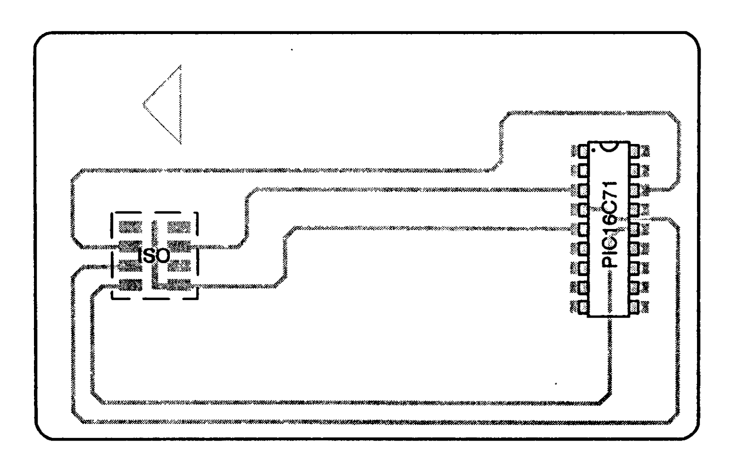


Рис. 3.15. Схема размещения элементов «фальшивой карты» ISO

на рис. 3.17) – в соответствии с AFNOR, который обречен на постепенный выход из обращения, так как является «франко-французским».

Обратите внимание, что направление счета выводов меняется от одной версии к другой. Поэтому необходимо тщательно располагать панельку с цанговыми контактами, которую не рекомендуется использовать на карте с микроконтроллером.

Естественно, эти «фальшивые карты» должны выполняться из одностороннего фольгированного стеклотекстолита толщиной 0,8 мм,

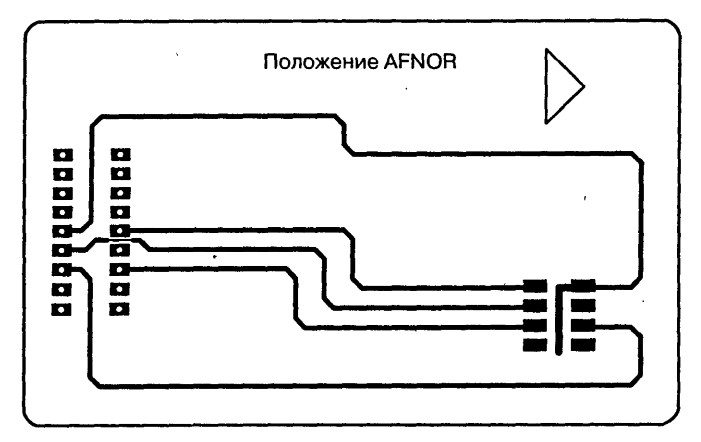


Рис. 3.16. Топология «фальшивой карты» AFNOR

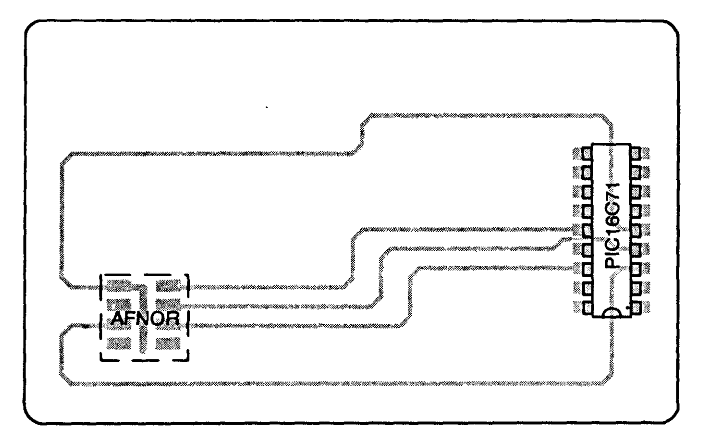


Рис. 3.17. Схема размещения элементов «фальшивой» карты AFNOR

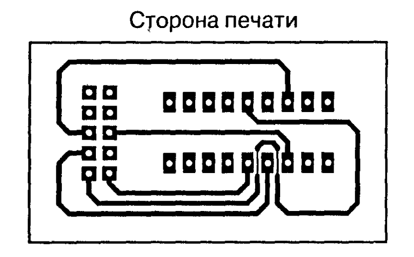


Рис. 3.18. Топология печатной платы и схема размещения элементов адаптера для PIC16CXX

который прекрасно подходит к картоприемникам, предусмотренным для карт толщиной 0,76 мм.

Вообще-то карты не вставляются в картоприемник настолько далеко, чтобы чип и панелька упирались в него. Однако, если есть сомнения, всегда можно немного удлинить плату имитатора карты. Во всяком случае, не стоит рассчитывать, что данная книга научит вас делать карты, которые сможет *проглотить* банкомат...

Читателям, которые уже создали набор рабочих инструментов, описанный в книге «Чип-карты. Устройство и применение в практических конструкциях», будет интересно изготовить из стеклотекстолита нормальной толщины совсем небольшую печатную плату, представленную на рис. 3.19.

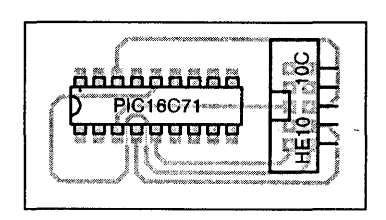


Рис. 3.19. Схема размещения элементов адаптера для PIC16CXX

Как видно по рис. 3.20, плата снабжена привычной разрезной колодкой с двумя рядами квадратных штырьков. С помощью кабеля с двумя разъемами HE10 плата может быть подключена к любой уже разработанной универсальной «фальшивой карте»: и ISO, и AFNOR, а также к SIM, совместимой с некоторыми типами мобильных телефонов GSM.

Настало время перейти к описанию программного обеспечения.

Код, который можно несколько помпезно назвать *маской «фальшивой карты»*, был разработан на языке ассемблера с помощью инструментария, представленного в наборе для самостоятельного изучения (kit) PICSTART 16B компании Microchip.

В действительности листинг является результатом ассемблирования (адреса ветвлений вычислены автоматически) исходного файла, который носит название PICPUCE.ASM и содержится на сайте www.dmk.ru.

```
16c5x/xx Cross-Assembler V4.12.01 Intermediate Fri Mar 24 11:13:44 1995 Page 1
Line  PC      Opcode
0001;       -- PICPUCE.ASM --
```

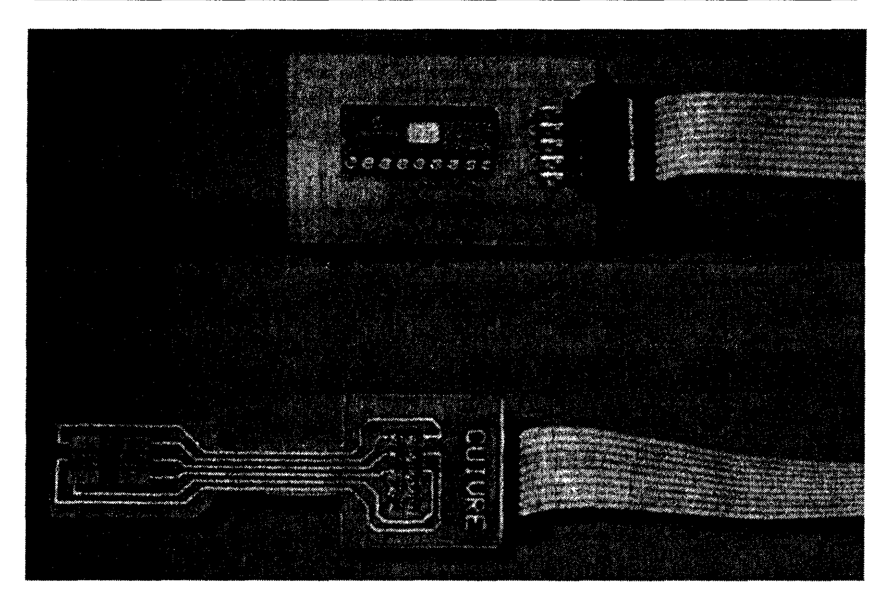


Рис. 3.20. Адаптер PIC16C71, подключенный к «фальшивой карте» SIM

```
;   Эмуляция асинхронной чип-карты на PIC16C71 или PIC16C84.
;   Copyright (c) 1995 Патрик Гелль.
;
;   Перемычки конфигурации: WDT и PuT не исп., генератор в режиме XT.
;
0000   0000   org 0
0000   2804          goto init
0000   0000   org 4
0004   1683   init   bsf 3,5         ;Инициализация порта А.
0005   1408          bsf 88,0        ;Режим порта ввода/вывода.
0006   1488          bsf 88,1        ;(CAN отключен.)
0007   1283          bsf 3,5
0008   1606          bsf 6,4
0009   2066          call tx         ;Вывод 3 в режим вывода.
0010   3032          movlw.50
000B   0090          movwv 10
000C   20A2   tempo  call delay1     ;Задержка перед ответом на сброс.
000D   0B90          decfsz 10,1
000E   280C          goto tempo

000F   303F          movlw 3F        ;Выдача ответа на сброс.
0010   208F          call even       ;(Имитация банковской карты BO'.)
0011   3065          movlw 65
```

```
0025 0012   208F            call even     ;Выдача байта контроля четности.
0026 0013   3025            movlw 25
0027 0014   2095            call odd      ;Выдача байта контроля нечетности.
0028 0015   3008            movlw 08
0029 0016   2095            call odd
0030 0017   3031            movlw 31
0031 0018   2095            call odd
0032 0019   3004            movlw 04
0033 001A   2095            call odd
0034 001B   306C            movlw 6C
0035 001C   208F            call even
0036 001D   3090            movlw 90
0037 001E   208F            call even
0038 001F   3000            movlw 00
0039 0020   208F            call even
0040
0041 ;Программа ожидания конфиденциального кода ABCDh по любому адресу.
0042
0043 0021   206A            call rx       ;Вывод 3 в режиме входа.
0044 0022   207E    clas    call recv     ;Ожидание приема байта.
0045 0023   30BC            movlw 0BC     ;Принятый байт = BCh? (класс ISO
                                          ;BULL CP8).
0046 0024   0691            xorwf 11,1
0047 0025   1D03            btfss 3,2
0048 0026   28A7            goto error    ;В противном случае молчание карты.
0049 0027   207E    pin     call recv

16c5x/xx Cross-Assembler V4.12.01 Intermediate Fri Mar 24 11:13:44 1995 Page 2

Line PC     Opcode

0050 0028   3020            movlw 20      ;Принятый байт = 20h? (презентация
                                          ;кода).
0051 0029   0691            xorwf 11,1
0052 002A   1D03            btfss 3,2
0053 002B   28A7            goto error    ;В противном случае молчание карты.
0054 002C   207E            call recv     ;Не воспринимать первый байт адреса.
0055 002D   207E            call recv     ;Не воспринимать второй байт адреса.
0056 002E   207E            call recv     ;Ожидание приема байта.
0057 002F   3002            movlw 02      ;Длина кода = 2 байта.
0058 0030   0691            xorwf 11,1
0059 0031   1D03            btfss 3,2
0060 0032   28A7            goto error    ;В противном случае молчание карты.
0061 0033   2066            call tx
0062 0034   20A2            call delay1
0063 0035   20A2            call delay1
0064 0036   3020            movlw 20      ;Выдача байта процедуры "без Vpp".
0065 0037   2095            call odd
```

```
0066  0038  206A          call rx
0067  0039  207E   cla    call recv
0068  003A  30AB          movlw 0AB      ;Первый представленный байт чипа =
                                         ;Abh?
0069  003B  0691          xorwf 11,1
0070  003C  1D03          btfss 3,2
0071  003D  28A7          goto error     ;В противном случае молчание карты.
0072  003E  207E          call recv
0073  003F  30CD          movlw 0CD      ;Второй представленный байт чипа =
                                         ;CDh?
0074  0040  0691          xorwf 11,1
0075  0041  1D03          btfss 3,2
0076  0042  28A7          goto error     ;В противном случае молчание карты.
0077  0043  2066          call tx
0078  0044  20A2          call delay1
0079  0045  20A2          call delay1
0080  0046  3090          movlw 90       ;Выдача протокола "код правильный".
0081  0047  208F          call even
0082  0048  3000          movlw 00
0083  0049  208F          call even
0084  004A  206A          call rx        ;Вывод 3 в режим ввода.
0085
0086  ;Программа ответа на требование чтения двух байт (по любому адресу).
0087
0088  004B  207E   class  call recv
0089  004C  30BC          movlw 0BC
0090  004D  0691          xorwf 11,1
0091  004E  1D03          btfss 3,2
0092  004F  28A7          goto error
0093  0050  207E   read   call recv
0094  0051  30B0          movlw 0B0      ;Принятый байт = B0h? (считывание).
0095  0052  0691          xorwf 11,1
0096  0053  1D03          btfss 3,2
0097  0054  28A7          goto error
0098  0055  207E   adr1   call recv      ;Не воспринимать первый байт адреса.
0099  0056  207E   adr2   call recv      ;Не воспринимать второй байт адреса.
0100  0057  207E   len    call recv      ;Не воспринимать байт длины.

16c5x/xx Cross-Assembler V4.12.01 Intermediate Fri Mar 24 11:13:44 1995 Page 3

Line  PC    Opcode

0101  0058  20A2   ack    call delay1
0102  0059  20A2          call delay1
0103  005A  2066          call tx        ;Вывод 3 в режим вывода.
0104  005B  30B0          movlw 0B0      ;Ответ карты.
```

```
005C   2095           call odd
005D   30CD   data1   movlw 0CD      ;Первый байт данных.
005E   2095           call odd       ;(Контроль нечетности.)
005F   30EF   data2   movlw 0EF      ;Второй байт данных.
0060   2095           call odd
0061   3090   me1     movlw 90       ;Протокол "выполнение правильное".
0062   208F           call even      ;(Контроль четности.)
0063   3000   me2     movlw 00
0064   208F           call even
0065   284B   loop    goto class

0066   1683   tx      bsf 3,5        ;Программа включения режима вывода.
0067   1205           bcf 85,4
0068   1283           bcf 3,5
0069   0008           return
006A   1683   rx      bsf 3,5        ;Программа включения режима ввода.
006B   1605           bsf 85,4
006C   1283           bcf 3,5
006D   0008           return
006E   008D   send    movwf 0D       ;Программа запуска УСАПП.
006F   098D           comf 0D,1
0070   3008           movlw 8
0071   008E           movwf 0E
0072   1205           bcf 5,4
0073   20A2           call delay1
0074   1003   next    bcf 3,0
0075   0D8D           rlf 0D,1
0076   1803           btfsc 3,0
0077   1605           bcf 5,4
0078   1C03           btfsc 3,0
0079   1205           bcf 5,4
007A   20A0           call delay2
007B   0B8E           decfsz 0E,1
007C   2874           goto next
007D   0008           return
007E   0191   recv    clrf 11        ;Стандартная программа приема УСАПП.
007F   1A05           btfsc 5,4
0080   289D           goto delay3
0081   209B           call delay4
0082   3008           movlw 8
0083   0090           movwf 10
0084   1003   rnext   bcf 3,0
0085   0D91           rlf 11,1
0086   1A05           btfsc 5,4
0087   1411           bsf 11,0
```

```
16c5x/xx Cross-Assembler V4.12.01 Intermediate Fri Mar 24 11:13:44 1995 Page 4
Line  PC    Opcode
0150  0088  20A2            call delay1
0151  0089  0B90            decfsz 10,1
0152  008A  2884            goto rnext
0153  008B  209B    parity  call delay4     ;Не воспринимать бит контроля
                                            ;четности.
0154  008C  0991            comf 11,1       ;Бит в 1 = лог. 0
                                            ;(обратное соглашение).
0155  008D  0008            return
0156  008E  20A2            call delay1
0157  008F  206E    even    call send       ;Выдача четного байта.
0158  0090  1605            bsf 5,4
0159  0091  20A2            call delay1
0160  0092  1605            bsf 5,4
0161  0093  20A2            call delay1
0162  0094  0008            return
0163  0095  206E    odd     call send       ;Выдача нечетного байта.
0164  0096  1205            bcf 5,4
0165  0097  20A2            call delay1
0166  0098  1605            bsf 5,4
0167  0099  20A2            call delay1
0168  009A  0008            return
0169  009B  3022    delay4  movlw.34        ;Выдержка времени 1,25 бит.
0170  009C  28A3            goto time
0171  009D  300E    delay3  movlw.14        ;Выдержка времени 1,5 бит.
0172  009E  20A3            call time
0173  009F  287E            goto recv
0174  00A0  301B    delay2  movlw.27        ;Задержка 1 бит (104 ES
                                            ;на 9600 бод).
0175  00A1  28A3            goto time
0176  00A2  301C    delay1  movlw.28        ;Длительность одного старт-/стоп-
                                            ;бита.
0177  00A3  008F    time    movwf 0F        ;Цикл выдержки времени.
0178  00A4  0B8F    redo    decfsz 0F,1
0179  00A5  28A4            goto redo
0180  00A6  3400            retlw 0
0181  00A7  28A7    error   goto error      ;Цикл (молчание и блокировка карты).
0182  0000          end                     ;(Для повторного старта выполнить
                                            ;команду "сброс".)
```

В результате трансляции с ассемблера при помощи MPALC создается файл в формате *intel hex*, который называется PICPUCE.OBJ. Он приведен ниже, выложен на сайте www.dmk.ru и напрямую совместим с программатором PICSTART компании MICROCHIP.

```
:020000000428D2
:10000800831608148814831206166620323090006E
:10001800A220900B0C283F308F2065308F20253090
:10002800952008309520313095200430952060C308B
:100038008F2090308F2000308F206A207E20BC30A7
:100048009106031DA7287E202030910603 1DA728AE
:100058007E207E207E20023091060031DA728662080
:10006800A220A220203095206A207E20AB30910665
:100078000031DA7287E20CD3091060 31DA7286620E2
:10008800A220A22090308F2000308F206A207E206E
:10009800BC309106031DA7287E20B0309106031DB1
:1000A800A7287E207E207E20A220A2206620B030B5
:1000B8009520CD309520EF30952090308F2000305E
:1000C8008F204B2883160512831208008316051605
:1000D80083120800 8D008D0908308E000512A220B9
:1000E80003108D0D0318051603 1C0512A0208E0B96
:1000F800742808009101051A9D289B20083090005B
:1001080003 10910D051A1114A220900B84289B202E
:1001180009 1090800A2206E200516A2200516A2202B
:1001280008006E200512A2200516A220080022 3021
:10013800A3280E30A3207E281B30A3281C308F0054
:080148008F0BA4280034A72846
:00000001FF
```

Файл может быть «зашит» в микросхемы PIC16C71 (версия с УФ-стиранием) или PIC16C84 (с технологией ЭСППЗУ), так как демонстрационные программы не используют память данных на ЭСППЗУ последнего.

Напротив, совершенно необходимо нейтрализовать сторожевой таймер-счетчик (WDT) и устройство выдержки времени включения напряжения (PUT), а также установить тактовый генератор в режим XT. Это выполняется посредством программирования перемычек в меню программного обеспечения MPSTART, копия содержимого экрана которого представлена на рис. 3.22.

Конечно, тем исследователям, которые пожелают продолжить эксперименты (под личную ответственность), создавая собственные прикладные программы, будет интересно заняться чтением и записью в энергонезависимой памяти, сохраняющей свои данные, когда карта вынимается из считывающего устройства.

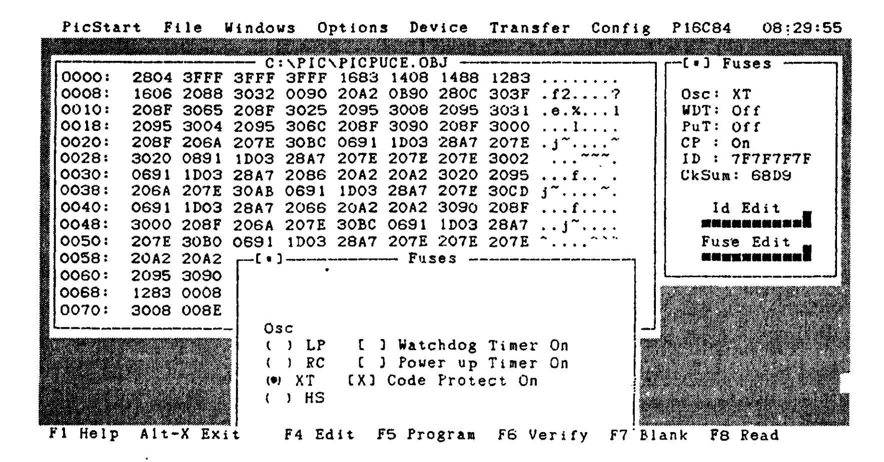


Рис. 3.21. Подготовка к программированию микроконтроллера с помощью программы MPSTART

Рис. 3.22. Устройство программирования PICSTART производства компании Microchip

Такое программное обеспечение содержит все основные программы, обычно отрегулированные с точностью до одного цикла тактового генератора, что позволяет выдавать четкий ответ на сброс, принимать и отправлять байты, переключать карту в режим молчания. Из этого режима ее может вывести лишь отключение от напряжения питания или полный сброс (это широко распространенный метод обеспечения безопасности, применяемый в «настоящих» картах на микропроцессоре).

Данная маска содержит также целую серию модулей для выполнения наиболее простых операций, начиная с двух программ, которые выполняют функции УСАПП для организации связи по последовательному каналу.

УСАПП, выполненный программно, не считается неполноценным по сравнению с его аппаратной реализацией. Подобное решение часто встречается в мире чип-карт, так как в этом случае более экономно используется кремний: считается, что поверхность, занимаемая УСАППом на кристалле, сравнима с поверхностью, которая может вместить 1500 байт ПЗУ, то есть практически полностью занять память микроконтроллера.

В ходе работы возникла проблема обработки бита контроля четности, предусмотренного стандартом. Система команд микроконтроллеров не позволяет реализовать его простым способом. Автор принял решение (как на GWBASIC) полностью игнорировать при приеме бит четности, а также сигнал ошибки, который на практике появляется лишь в исключительных случаях. Как следствие, в программе предусмотрен вызов операции выдержки времени, равной длительности бита, который требуется «перескочить».

В момент передачи, наоборот, «перепрыгивание» через бит, который контролируется промышленными устройствами чтения-записи, не требуется.

Имея в виду, что в рамках проводимых операций, носящих в основном экспериментальный характер, заранее известны байты, предназначенные для передачи, автор предусмотрел две различные процедуры связи. Одна из них (метка *even*) принудительно включает бит четности в 1, а другая (метка *odd*) – в 0. Таким образом, производят либо одну, либо другую процедуру в зависимости от того, какое число передается: четное или нечетное.

Было бы неплохо разработать программу автоматического вычисления бита четности, но для оставшихся ресурсов памяти можно найти и лучшее применение.

Настало время обратиться к примерам на основе базовых программ. Эти коды должны выполняться по порядку при испытаниях «фальшивой карты» на устройстве чтения-записи (рис. 2.4), которое управляется с помощью программы INVERSE.BAS.

В первом случае карта должна выдать ответ на сброс, полностью идентичный ответу, который выдают наиболее современные банковские карты (CP8 M4 B0' с памятью типа ЭСППЗУ):

```
3F 65 25 08 31 04 6C90 00
```

Здесь нет никакой провокации, это лишь удобное средство для оповещения считывающего устройства о том, что карта работает по обратному соглашению и ей достаточно напряжения программирования 5 В. К тому же не приходится прибегать к программированию ответа на сброс по размерам, вызывающему определенные сложности.

Во втором примере карте присваивается класс ISO BCh (это класс карт BULL CP8) и программируется ожидание представления конфиденциального кода на два байта (ABCDh) по любому адресу.

Пользуясь случаем, необходимо отметить, что в соответствии со стандартом ISO устройство чтения-записи в первую очередь выдает блок из пяти байт, который называется *заголовком*, и предлагает представить код:

- BCh (класс ISO карты);
- 20h (код операции «представление кода»);
- первый байт адреса (старшие значащие разряды);
- второй байт адреса (младшие значащие разряды);
- байт, уточняющий длину кода (в данном случае 2 байта, то есть 02h).

Карта должна немедленно ответить *байтом процедуры*, идентичным коду операции из системы представления кода (20h). Если карта дает ответ 21h (младший бит в состоянии лог. 1), значит, она требует напряжения программирования Vpp. Только после получения этого байта считывающее устройство отправляет два байта конфиденциального кода, на которые карта отвечает двумя байтами 90h и 00h, – характерный вид протокола успешного завершения операции.

Отмечается, что при малейшей ошибке (неверный класс ISO, код операции отличный от 20h, ложный код и т.п.) программа «зацикливается» на блокировку карты в состоянии полного молчания (метка *ошибка*). «Настоящая» карта скорее выдаст протокол, уточняющий природу происшедшего сбоя, и читатели смогут потренироваться в программировании такой операции (это очень просто).

Теперь покажем, как запрос программы, без использования байта ее возврата, позволяет «перепрыгнуть» через байт, который не нужен. В данном примере речь идет о двух байтах, уточняющих адрес, по которому должен быть представлен конфиденциальный код.

В большинстве случаев адрес кода пользователя (PIN-кода) будет 0000h, но для решения задачи подходит любой. Только если был представлен правильный код, следующая команда может быть выполнена, причем неограниченное число раз до тех пор, пока не возникнет ошибка и программа перейдет в состояние молчания.

В третьем примере карта программируется на выдачу обоих байтов CDh и Efh, за которыми следует протокол 90 00, в ответ на каждое требование считывания двух байт (код операции B0h) по любому адресу:

```
BCh; B0h; ADR1; ADR2; 02h
```

Неплохое упражнение – указать единственный адрес, по которому эти данные могли бы быть прочитаны (достаточно добавить восемь команд).

Имея микроконтроллеры, защищенные от повторного считывания их программы, можно достичь при очень малых затратах высочайшей безопасности данных, встроенных в само тело программного обеспечения и, следовательно, в память ППЗУ.

Для того чтобы прочитать эти два байта, необходимо представить конфиденциальный код из двух байт (256×256, то есть возможны 65536 комбинаций) и, желательно, указать один адрес, также выраженный в двух байтах, даже если еще ничего не предусмотрено для запоминания (в байте ЭСППЗУ) числа попыток представления фальшивого кода и окончательной блокировки карты после трех неудачных попыток. С помощью PIC16C84 это реализуется удивительно просто.

Поскольку данная глава посвящена безопасности, стоит провести объективное сравнение между экспериментальной и обычной пластиковой картой на микропроцессоре.

Ясно, что представленная операционная мини-система, даже если она в состоянии воспроизвести основы «поведения» любой асинхронной карты, не идет ни в какое сравнение с маской карты COS или CP8. «Настоящая» карта на микропроцессоре наряду со значительно бóльшим объемом памяти располагает ресурсами, обеспечивающими безопасность, которые основываются, в частности, на алгоритмах кодирования, позволяющих препятствовать перехвату и последующему воспроизведению диалога между картой и устройством чтения-записи.

Лишенная таких возможностей чип-карта, разработанная в данной книге, является в определенной степени копируемой. Однако должна исключаться вероятность ее использования для копирования надежно защищенных приложений.

Во многих странах мира одновременно осуществлялось внедрение чип-карт, и появилась надежда, что это приведет к всеобщей унификации. Но микроэлектроника продолжает развиваться, и теперь нужно принимать в расчет T2G (французские телефонные карты второго поколения) и Eurochip, разработанные на базе немецких Telefonkarte.

Наряду со старой n-МОП ЭППЗУ-технологией первых французских телефонных карт (отныне их надо называть T1G), которые все еще продолжают «нести вахту», существуют уже по крайней мере две ЭСППЗУ-технологии, которые только и ждут, когда ими заинтересуются...

МИНИ-УСТРОЙСТВО ЧТЕНИЯ-ЗАПИСИ ISO/AFNOR

Хотя запись в телефонные карты (предпочтительно в те, у которых не осталось единиц услуг) – очень увлекательное занятие, существует множество интереснейших приложений, где используется только чтение.

Освобожденное от схемы, служащей для управления напряжением программирования Vpp, необходимым для записи в карты с памятью ППЗУ, устройство чтения синхронных чип-карт представляет собой чрезвычайно простое приспособление. В данном случае оно питается от обычной батарейки.

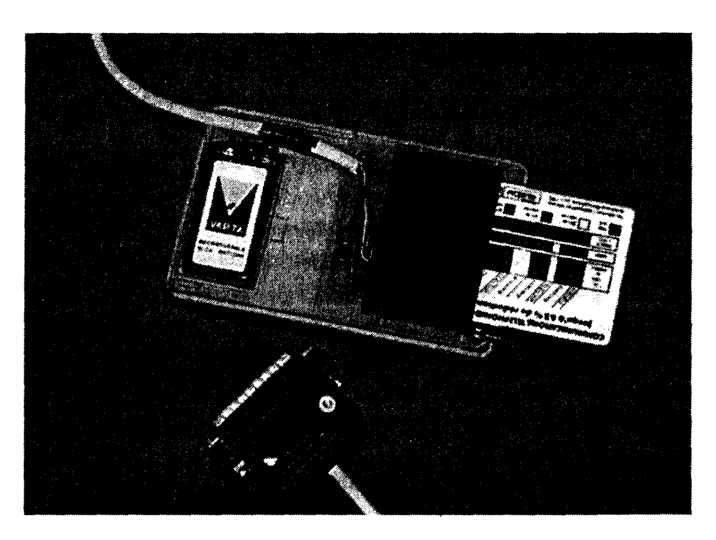


Рис. 4.1. Устройство чтения для отработанной синхронной карты

Но подобное устройство чтения способно также записывать информацию в карты второго поколения, которым вполне достаточно одного напряжения питания 5 В для работы ЭСППЗУ.

Схема, представленная на рис. 4.2, полностью совместима с программным обеспечением, опубликованным в книгах «Чип-карты». Устройство и применение в практических конструкциях» и «Как превратить персональный компьютер в универсальный программатор».

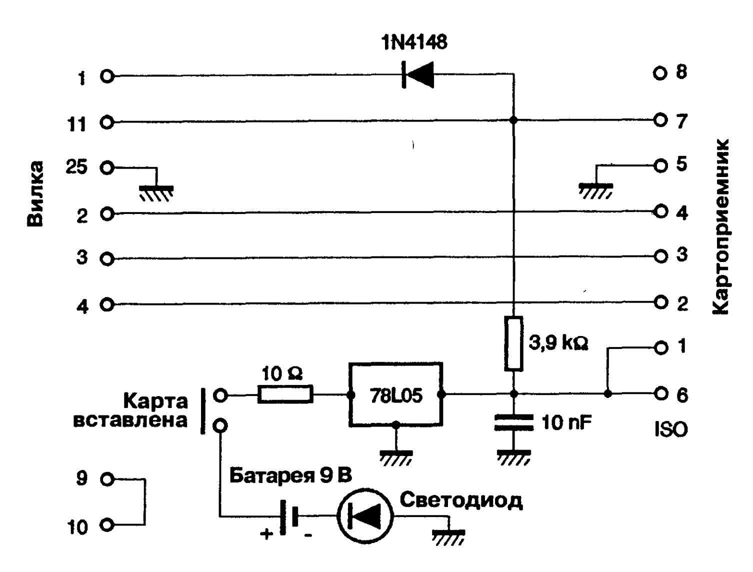


Рис. 4.2. Схема устройства чтения синхронных карт

Тем не менее она претерпела некоторые немаловажные изменения.

Соединение контактов 9 и 10 разъема DB25 типа «вилка» позволяет наиболее совершенному программному обеспечению автоматически распознавать параллельный порт, к которому подключено устройство: LPT1 или LPT2. При минимальных затратах к компьютеру, имеющему еще один свободный слот, подключается второй порт Centronics. Это дает возможность одновременно пользоваться услугами считывающего устройства и принтера, обходясь без утомительного переключения кабелей.

Благодаря простому включению диода между контактами 1 и 11 того же самого разъема устройство чтения совместимо с двунаправленными линиями данных некоторых типов карт, например работающих по протоколу I^2C.

В топологии печатной платы, изображенной на рис. 4.3, используются две пары контактов картоприемника (контактные гребенки ISO и AFNOR включены параллельно).

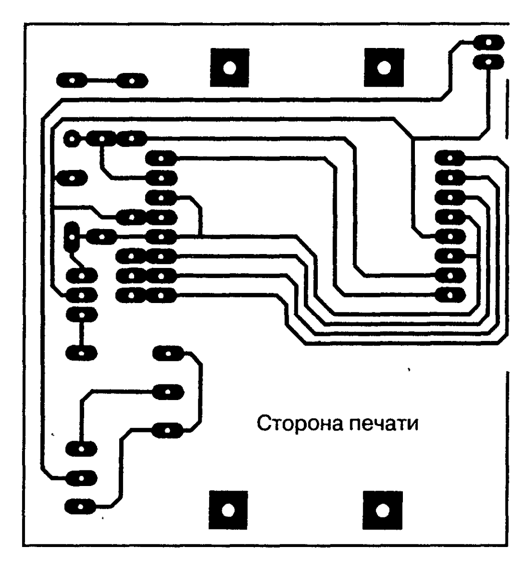


Рис. 4.3. Печатная плата устройства чтения синхронных карт

Все элементы источника питания на 5 В, размещенные в соответствии с рис. 4.4, были оптимизированы таким образом, чтобы обеспечить длительную автономию с простой щелочной батареей напряжением 9 В (приблизительно 100 часов присутствия карты в считывающем устройстве).

Размеры этой конструкции позволяют поместить ее в большинстве пластиковых корпусов типа «калькулятор» с предусмотренным для батареи местом, где для вставки карты необходимо сделать щель напротив картоприемника, одно отверстие для светодиода, свидетельствующего о включении питания, и второе – для соединительного кабеля связи с ПК.

На сайте издательства «ДМК» www.dmk.ru можно найти программное обеспечение для большинства наиболее распространенных карт, совместимое с данным устройством чтения. Необходимо инсталлировать программу CARTES.EXE с сервисной программой INSTALL.EXE, а также использовать любую телефонную карту в роли «логического ключа». Кроме этого, предусмотрена небольшая программа, которая может рассматриваться как необходимый минимум.

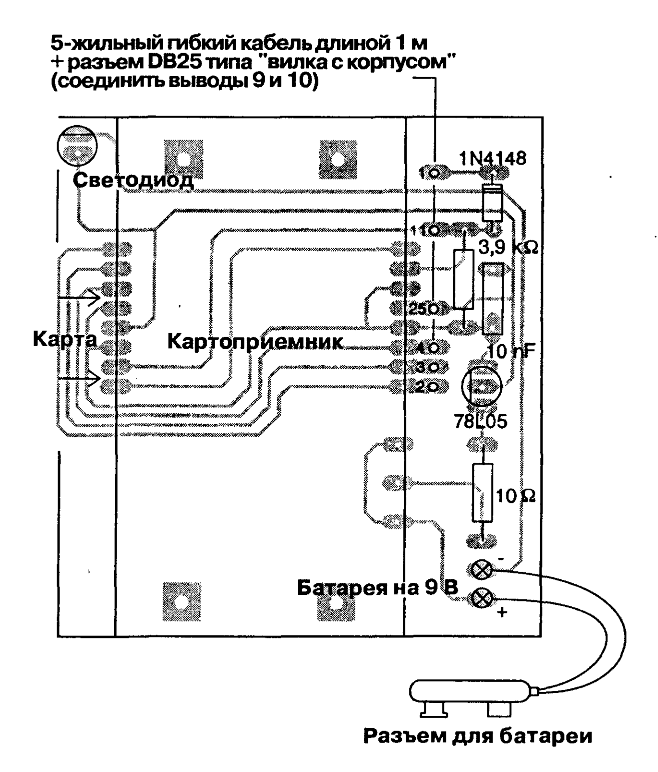


Рис. 4.4. Схема размещения элементов на плате устройства чтения синхронных карт

Ее преимущество заключается в модульности, открывающей широкие возможности будущего развития системы:

```
10 REM -- MINILECT.BAS --
20 KEY OFF:CLS:DEF SEG=0
30 S2=PEEK(&H40A)+256*PEEK(&H40B) 'для LPT2:
40 S1=PEEK(&H408)+256*PEEK(&H409) 'для LPT1:
50 OUT S2,0:E2=S2+1
60 IF (INP(E2) AND 64) <> 0 THEN S=S1:GOTO 100
70 OUT S2,128
80 IF (INP(E2) AND 64) <> 64 THEN S=S1:GOTO 100
90 S=S2
100 E=S+1
110 DIM B$(256)
120 KEY OFF
130 CLS
140 OUT S,0
150 PRINT"Вставить карту, затем нажать ENTER"
```

```
160 INPUT Z$
170 CLS
180 OUT S,250:OUT S,248
190 FOR I=1 TO 256
200 OUT S,249
210 D=INP(E):D= (D AND 128)
220 IF D=128 THEN B$(I)="0"
230 IF D<>128 THEN B$(I)="1"
240 OUT S,251
250 NEXT I
260 N=1
270 FOR F=1 TO 8
280 FOR G=1 TO 8
290 FOR H=1 TO 4
300 PRINT B$(N);:N=N+1
310 NEXT H
320 PRINT" ";:NEXT G
330 PRINT:NEXT F
340 PRINT:PRINT:PRINT"Для сохранения файла набрать его имя, затем нажать ENTER"
350 PRINT:PRINT"Не сохранять: заменить карту и нажать ENTER"
360 PRINT:PRINT:INPUT Z$
370 IF Z$="" THEN 170
380 FOR F=1 TO LEN(Z$)
390 IF MID$(Z$,F,1)="." THEN 420
400 NEXT F
410 Z$=Z$+".CAR"
420 OPEN Z$ FOR OUTPUT AS #1
430 N=1
440 FOR F=1 TO 8
450 FOR G=1 TO 8
460 FOR H=1 TO 4
470 PRINT#1, B$(N)+" ";:N=N+1
480 NEXT H
490 PRINT#1," ";:NEXT G
500 PRINT#1,:NEXT F
510 CLOSE#1:GOTO 130
520 REM (c)1995 Patrick GUEULLE
```

На самом деле программа MINILECT.BAS читает 256 последовательных бит на предварительно обнуленной карте (после сигнала «сброс»), однако может сохранить эти данные на диске в формате CAR. Таким образом, совместимость обеспечена!

Программа MINILECT.BAS в приведенной выше версии работает с устройством чтения через параллельный порт LPT1, для использования порта LPT2 достаточно стереть ее строку 40.

```
10 REM -- T1G.BAS --
20 DIM N(256):DIM M(256)
30 KEY OFF:CLS
40 PRINT:GOTO 350
50 CLS:I=0
60 FOR F=1 TO 8
70 FOR G=1 TO 8
80 FOR H=1 TO 4
90 I=I+1
100 N$="1":IF N(I)=0 THEN N$="0"
110 PRINT N$;
120 NEXT H
130 PRINT" ";:NEXT G
140 PRINT
150 IF F=3 THEN PRINT
160 NEXT F:PRINT
170 PRINT"CHOIX + ENTER:":PRINT
180 PRINT"0 -> возврат в DOS (выход)"
190 PRINT"1 -> загрузка"
200 PRINT"2 -> анализ"
210 PRINT"3 -> отображение в шестнадцатеричном коде"
220 PRINT"4 -> отображение в двоичном коде"
230 PRINT"5 -> контроль четности"
240 PRINT"6 -> вызов DOS (временно)"
250 INPUT Z$
260 IF Z$="0" THEN SYSTEM
270 IF Z$="6" THEN SHELL:CLS:GOTO 170
280 IF Z$="1" THEN 340
290 IF Z$="2" THEN 480
300 IF Z$="4" THEN 50
310 IF Z$="5" THEN 900
320 IF Z$="3" THEN 1220
330 GOTO 250
340 CLS
350 PRINT" Имя загружаемого файла"
360 INPUT S$
370 IF S$="" THEN 340
380 FOR F=1 TO LEN(S$)
390 IF MID$(S$,F,1)="." THEN 420
400 NEXT F
410 S$=S$+".CAR"
420 OPEN S$ FOR INPUT AS #1
430 CLS:PRINT"-- Идет загрузка --"
440 FOR F=1 TO 256
450 INPUT#1,Q: N(F)=Q
460 NEXT F
```

```
470 CLOSE#1:GOTO 50
480 CLS:PRINT"Код семейства: ";
490 F$="":A=9
500 GOSUB 1130:F$=F$+K$
510 A=13:GOSUB 1130:F$=F$+K$
520 IF N(9)=0 THEN 550
530 PRINT:PRINT"Телефонная карта не опознана"
540 PRINT:GOTO 170
550 PRINT" (Телефонная карта)"
560 PRINT"Номер серии: ";
570 FOR A=17 TO 29 STEP 4
580 GOSUB 1130:NEXT A
590 FOR A=41 TO 53 STEP 4
600 GOSUB 1130:NEXT A
610 PRINT:PRINT"Сообщение идентичности: ";
620 A=57:GOSUB 1130
630 A=61:GOSUB 1130
640 A=73:GOSUB 1130
650 A=77:GOSUB 1130
660 PRINT:PRINT"Параметры программирования: ";
670 A=81:GOSUB 1130
680 IF K$="0" THEN PRINT" (50 ms / 25 V)"
690 IF K$="1" THEN PRINT" (50 ms / 21 V)"
700 PRINT"Служебный код: ";
710 A=85:GOSUB 1130
720 IF K$="0" THEN PRINT" (Карта израсходована)";
730 PRINT:PRINT"Финансовые возможности: ";
740 P$=""
750 A=89:GOSUB 1130:P$=P$+K$
760 A=93:GOSUB 1130:P$=P$+K$
770 P=VAL(P$):P=(10*P)-10
780 PRINT" (";P;" единиц)"
790 PRINT"Использовано: ";
800 C=0
810 FOR F=97 TO 248
820 IF N(F)=1 THEN C=C+1
830 NEXT F:C=C-10
840 PRINT C;" UTC"
850 IF C<P THEN PRINT"Остается: ";P-C;" UTC":BEEP
860 IF C>P THEN PRINT"Карта испорчена":BEEP
870 IF C=P THEN PRINT" Кредит исчерпан"
880 PRINT:PRINT:GOTO 170
890 GOTO 170
900 CLS:PRINT"Контроль четности:":PRINT
910 IF N(9)=0 THEN 950
920 PRINT"Без значения!"
```

```
930 PRINT"(Только телефонные карты)":PRINT
940 GOTO 170
950 W=1:GOSUB 990
960 W=2:GOSUB 990
970 W=3:GOSUB 990
980 PRINT:GOTO 170
990 PRINT"БЛОК №";W;": ";
1000 K=0
1010 X=(32*W)-31
1020 IF N(X)=0 THEN 1110
1030 FOR F=0 TO 4
1040 X=X+1:IF N(X)=1 THEN K=K+2^(4-F)
1050 NEXT F
1060 Z=0
1070 FOR F=X+1 TO X+26
1080 IF N(F)=0 THEN Z=Z+1
1090 NEXT F
1100 IF Z=K THEN PRINT"Четность правильна":RETURN
1110 PRINT"Четность неправильна":RETURN
1120 END
1130 K=0
1140 FOR J=0 TO 3
1150 B=N(A+J)
1160 IF B=1 THEN K=K+2^(3-J)
1170 NEXT J
1180 IF K<10 THEN K$=CHR$(48+K)
1190 IF K>=10 THEN K$=CHR$(55+K)
1200 PRINT K$;
1210 RETURN
1220 CLS
1230 FOR L=1 TO 225 STEP 32
1240 FOR N=0 TO 28 STEP 4
1250 A=L+N:GOSUB 1130
1260 NEXT N
1270 PRINT:NEXT L
1280 PRINT
1290 GOTO 170
1300 END
1310 C$="(c)1994 Patrick GUEULLE"
```

Программа `T1G.BAS` предназначена исключительно для французских телефонных карт первого поколения (код семейства начинается с нуля). Однако она может использоваться для декодирования файлов, которые считаны с карт при помощи программы `MINILECT.BAS`, для того чтобы извлечь максимум информации. Далее будет показано, как выгладит на экране результат работы программы:

```
1101 0011 0000 0100 0000 0000 0001 0110
1100 1111 0010 0000 0110 1000 0001 0000
1100 0111 1000 1011 0001 0000 0000 0110
1111 1111 1111 1111 1111 1111 1111 1111
1111 1111 1111 1111 1111 1111 1111 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 1111 1111
Номер + ENTER:
0 -> возврат в DOS (выход)
1 -> загрузка
2 -> анализ
3 -> отображение в шестнадцатеричном коде
4 -> отображение в двоичном коде
5 -> контроль четности
6 -> вызов DOS (временно)
?
Код семейства: 04 (Телекарта)
Номер серии: 00162068
Сообщение идентичности: 108B
Параметры программирования: 1 ( 50 ms / 21 V )
Служебный код: 0 (одноразовая карта)
Финансовые возможности: 06 ( 50 единиц)
Использовано: 50 UTC
Кредит исчерпан
Контроль четности:
BLOC N° 1: четность правильна
BLOC N° 2: четность правильна
BLOC N° 3: четность правильна
D3040016
CF206810
C78B1006
FFFFFFFF
FFFFFFF0
00000000
00000000
000000FF
```

Можно отметить, что «контроль четности» позволяет проверить правдоподобность содержимого «зоны изготовителя» (96 первых бит, защищенных известной «перемычкой»).

Один-единственный искаженный бит (который, вполне вероятно, присутствует на старых картах, более десяти лет хранившихся в плохих условиях) испортит весь тест.

Декодирование сообщения идентичности, со своей стороны, дополняет декодирование номера серии. В каком-то смысле он подтверждает происхождение чипа, так как рассчитывается по сверхсекретной

формуле. Отображение в шестнадцатеричном коде являет исключительно компактную картину, для некоторых более ясную, чем простые блоки из четырех бит.

Далее в этой главе будет показан эквивалент данной программы для карт T2G, а также для EUROCHIP. Кроме того, не представляет большого труда разработать отдельные версии и для других семейств карт (например, для иностранных телефонных карт с ППЗУ).

На сайте www.dmk.ru содержится целый набор примеров реальных файлов в формате CAR, позволяющих начать проведение экспериментов без считывающего устройства и даже без самой карты.

РАСПОЗНАВАНИЕ ЧИПОВ С ПОМОЩЬЮ СПЕЦИАЛЬНОЙ ПРОГРАММЫ

Несмотря на то что коллекционеры телефонных карточек привычно называют «чипом» микромодуль, такое наименование ошибочно. Настоящий электронный *чип* – это та самая память, где хранятся единицы услуг.

Маленький кремниевый квадрат со стороной размером приблизительно 1 мм, подлинное «сердце» чип-карты, приклеен с задней стороны гибкой подложки, на которой расположена группа из шести-восьми контактов. Кристалл, утопленный в толщине пластмассы карты, чаще всего остается скрытым. Его можно увидеть на обратной стороне карт, имеющих прозрачную, коричневую или белую подложку, но не на картах с черной подложкой или совсем без нее.

На некоторых образцах, например изготовленных компанией Schlumberger, пластмасса прозрачна и достаточно чиста, что позволяет довольно четко рассмотреть чип. Похоже, не все телефонные карты имеют одинаковую модель кремниевого чипа. Кропотливые исследования позволили автору выявить по крайней мере три возможных варианта.

Наиболее ранний легко распознать по буквам ET высотой 0,05 мм, которые написаны в углу темной обширной площадки, видимой невооруженным глазом. Но для того, чтобы прочитать дату маски (1983), вытравленную в центре чипа ET 1001, нужен микроскоп, так как цифры имеют в высоту едва ли 20 микрон, то есть не составляют даже 0,02 мм.

Более поздняя и конкурентноспособная версия компании Texas Instruments имеет дату маски 1986. Чуть меньшая по размерам, чем чип ET 1001, она при просмотре под микроскопом легко идентифицируется по логотипу марки (черепицеобразные буквы T и I, выгравированные в углу).

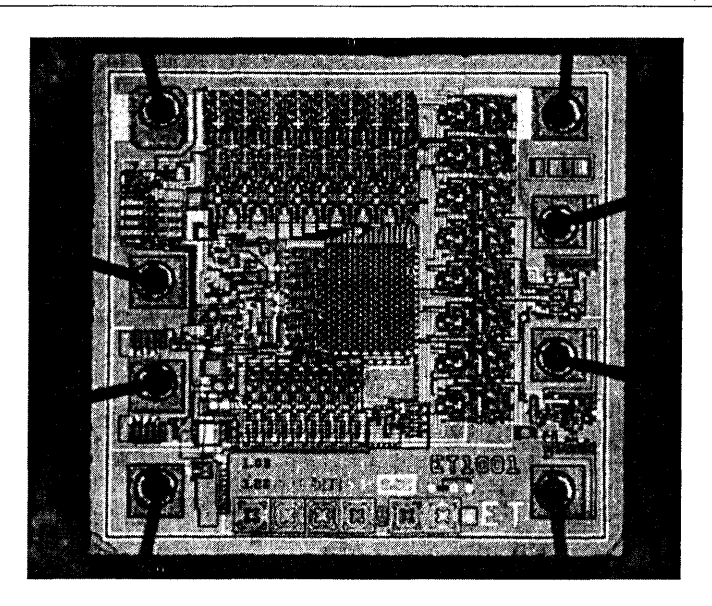


Рис. 4.5. Чип телефонной карты ET 1001 (вид в микроскоп)

Этот кристалл часто встречается в некоторых сериях телефонных карт марки Schlumberger или Solaic, но ни разу не замечен в картах Gemplus. Возможно, это обусловлено историческими связями фирмы с компанией SGS-Thomson.

Наиболее распространенным чипом, которым снабжено большинство телефонных карт, выпускаемых в настоящее время (и который будет вытеснен разве что стандартом T2G), является TS 1001 компании SGS-Thomson; дата маски – 1987. Он самый маленький из трех, а потому, без сомнения, самый экономичный в производстве, поскольку кремний стоит дорого. Отмечается его высокое сходство с ET 1001, вся пустовавшая площадь которого удалена.

Интересно отметить, что чипы марок Texas и SGS-Thomson поразному реагируют на операции считывания. Чипы Texas показывают прочитанный бит на контакте ISO7 сразу после нарастающего фронта тактового импульса, в то время как чипы SGS-Thomson ждут возвращения тактового сигнала в нуль. Кроме этого, чип производства SGS-Thomson может быть прочитан только при наличии напряжения программирования Vpp, составляющего по крайней мере 5 В, в то время как чип Texas Instruments переносит и напряжение равное нулю.

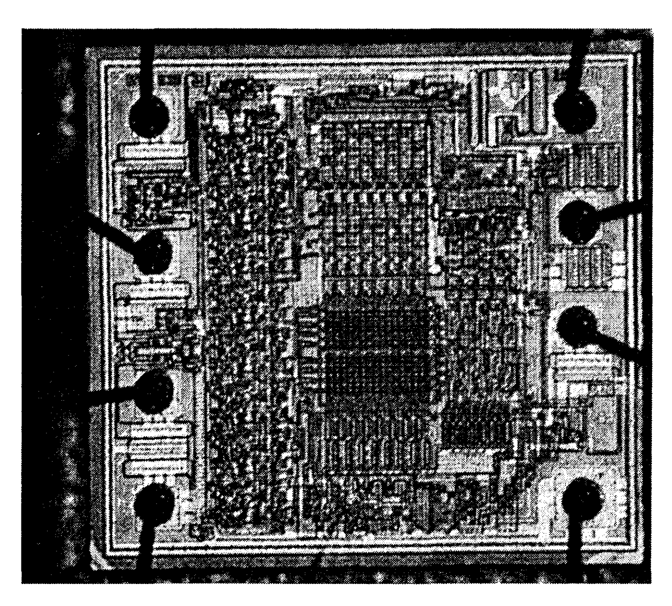


Рис. 4.6. Чип телефонной карты TMS 3561 (вид в микроскоп)

В связи с этим была предпринята разработка небольшого алгоритма распознавания чипов, представленного ниже. Программа называется TEXAS.BAS.

```
10 REM -- TEXAS --
20 DEF SEG=0:KEY OFF:CLS
30 S2=PEEK(&H40A)+256*PEEK(&H40B) 'для LPT2:
40 S1=PEEK(&H408)+256*PEEK(&H409) 'для LPT1:
50 OUT S2,0:E2=S2+1
60 IF (INP(E2) AND 64) <> 0 THEN S=S1:GOTO 100
70 OUT S2,128
80 IF (INP(E2) AND 64) <> 64 THEN S=S1:GOTO 100
90 S=S2
100 E=S+1
110 UC=0:UD=0:ZC=0:ZD=0:OUT S,0
120 PRINT"Вставить карту, затем нажать ENTER"
130 INPUT Z$:CLS
140 OUT S,250:OUT S,248
150 FOR F=1 TO 16
160 D=INP(E):D= (D AND 128)
170 IF D=128 THEN ZD=ZD+1
180 IF D<>128 THEN UD=UD+1
190 OUT S,249
```

```
200 C=INP(E):C= (C AND 128)
210 IF C=128 THEN ZC=ZC+1
220 IF C<>128 THEN UC=UC+1
230 OUT S,251
240 NEXT F:PRINT
250 IF ZC=0 OR UC=0 THEN PRINT "Испорченная телефонная карта":GOTO 280260 IF
ZC=ZD THEN PRINT "Чип TMS3561 (Texas) или ЭСППЗУ"
270 IF UD=16 AND UC<16 THEN PRINT "Чип ET1001 или TS1001 (Thomson)"
280 PRINT:PRINT:GOTO 110
290 REM (c)1995 Patrick GUEULLE
```

Эта программа также включена в CARTES.EXE, поскольку представляет безусловный интерес для коллекционеров: наличие того или иного чипа внутри некоторых типов телефонных карт порой считается исключительной редкостью.

ТЕЛЕФОННЫЕ КАРТЫ И ЗАЩИТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Описываемая в книге программа CARTES.EXE имеет защиту от копирования на базе телефонной карты-ключа. Это наглядно показывает, чего можно добиться с помощью очень простых средств.

Любое лицо, располагающее сервисной программой INSTALL.EXE, может сделать столько копий, сколько имеется телефонных карт, которые смогут служить ключом. Напротив, программа CARTES.EXE, инсталлированная на дискету или жесткий диск, не содержащий INSTALL.EXE, может быть запущена только в присутствии той единственной телефонной карты-ключа, которая использовалась при установке.

По такому принципу (но по несколько отличному алгоритму) каждый может защитить свои собственные программы, компилируемые посредством транслятора TURBO-BASIC, с помощью служебной программы PROTECT.BAS:

```
10 REM -- PROTECT.BAS --
20 DEF SEG=0:KEY OFF
30 S1=PEEK(&H408)+256*PEEK(&H409)
40 S2=PEEK(&H40A)+256*PEEK(&H40B)
50 OUT S2,0:E2=S2+1
60 IF (INP(E2) AND 64) <> 0 THEN S=S1:GOTO 100
70 OUT S2,128
80 IF (INP(E2) AND 64) <> 64 THEN S=S1:GOTO 100
90 S=S2
100 E=S+1
110 DIM N(256):DIM M(256)
120 CLS:PRINT"Защита программного обеспечения с помощью телефонной карты"
```

```
130 PRINT"===================================="
140 PRINT:PRINT
150 IF S=S2 THEN PRINT" (Устройство чтения-записи на LPT2)"
160 IF S=S1 THEN PRINT" (Устройство чтения-записи на LPT1)"
170 PRINT:PRINT"Вставить карту-ключ, затем нажать ENTER"
180 INPUT Z$:CLS:KK$=""
190 OUT S,250:OUT S,248
200 FOR F=1 TO 12
210 K=0
220 FOR G=0 TO 7
230 OUT S,249
240 D=INP(E):D=(D AND 128)
250 IF D<>128 THEN K=K+2^(7-G)
260 OUT S,251:NEXT G
270 K=K XOR 55
280 KK$=KK$+CHR$(K):NEXT F
290 OUT S,0
300 OUT S,250:OUT S,248
310 OPEN "1995.(c)" FOR OUTPUT AS #3
320 KK$=KK$+" (c)1995 Patrick GUEULLE"
330 PRINT #3,KK$
340 CLOSE #3
350 PRINT KK$:PRINT:PRINT
360 END
370 REM (c)1995 Patrick GUEULLE
380 RETURN
```

Порядок действий прост: необходимо вставить модуль защиты PROTEGE.BAS в исходный текст, написанный на GWBASIC, той программы, которую надо защитить. Необходимо это сделать таким образом, чтобы по команде RUN он выполнялся первым.

```
1 REM -- PROTEGE.BAS --
2 DEF SEG=0:KEY OFF
3 S1=PEEK(&H408)+256*PEEK(&H409)
4 S2=PEEK(&H40A)+256*PEEK(&H40B)
5 OUT S2,0:E2=S2+1
6 IF (INP(E2) AND 64) <> 0 THEN S=S1:GOTO 15
7 OUT S2,128
8 IF (INP(E2) AND 64) <> 64 THEN S=S1:GOTO 15
9 S=S2
15 E=S+1
20 DIM N(256):DIM M(256)
30 CLS:PRINT"Программное обеспечение, защищенное с помощью телефонной карты"
40 PRINT"=============================="
41 PRINT" (c)1992,1995 Patrick GUEULLE":PRINT
45 IF S=S2 THEN PRINT" (Устройство чтения-записи на LPT2)"
46 IF S=S1 THEN PRINT" (Устройство чтения-записи на LPT1)"
```

```
47 GOSUB 5000
50 CLS:FOR F=1 TO 10
55 PRINT"Карта-ключ опознана, выполнение разрешено"
60 PRINT:NEXT F
4999 END
5000 PRINT:PRINT"Вставить карту-ключ, затем нажать ENTER"
5010 BEEP:INPUT Z$:CLS:KK$=""
5020 OUT S,250:OUT S,248
5030 FOR F=1 TO 15
5032 K=0
5035 FOR G=0 TO 7
5060 OUT S,249
5070 D=INP(E):D=(D AND 128)
5080 IF D<>128 THEN K=K+2^(7-G)
5090 OUT S,251:NEXT G
5091 K=K XOR 55
5099 KK$=KK$+CHR$(K):NEXT F
5100 OUT S,0
5150 OUT S,250:OUT S,248
5200 OPEN "1995.(c)" FOR INPUT AS #3
5210 INPUT#3,XX$
5220 IF LEFT$(XX$,12)=LEFT$(KK$,12) THEN RETURN
5230 CLS:PRINT:PRINT"Карта-ключ не опознана":PRINT:END
5250 CLOSE #3
5300 REM(c)1995 Patrick GUEULLE
5999 RETURN
```

Проделав это, следует откомпилировать программу при помощи TURBO-BASIC таким образом, чтобы получить исполняемый EXE-файл. Кстати, можно уменьшить его размер с помощью общедоступного программного обеспечения LZEXE или коммерческого динамического архиватора, например SHRINKER компании BLINKinc.

Та же самая программа может копироваться напрямую на любые дискеты, но не будет работать без «секретного файла», созданного PROTECT.BAS. Такой файл подготавливается (совместно с любым устройством чтения-записи) на базе телефонной карты, поставляемой в качестве ключа, или же самодельного аппаратного ключа, в который вмонтирован чип из использованной телекарты (см. книгу "Montages a composants programmables")[1]. Теперь надо представить, что программа, которую необходимо защитить, уже откомпилирована (например, PROTEGE.EXE) и размещена вместе с PROTECT.EXE (откомпилированная версия PROTECT.BAS) на дискете, помещенной в дисковод B.

[1] Русский перевод («Схемы с программируемыми компонентами») готовится к выпуску в издательстве «ДМК».

Чтобы создать защищенную версию на дискете, помещенной в дисковод А, необходимо последовательно выполнить нижеуказанные операции, разумеется, из приглашения DOS для диска А:

1. COPY B:PROTEGE.EXE
2. B:PROTECT

Сначала по запросу PROTECT.EXE в соответствующее устройство чтения-записи вставляется телефонная карта. Теперь ее следует хранить вместе с дискетой, с которой она связана и на которой создается дополнительный файл, называемый 1995.(c).

При каждом запуске защищенная программа требует введения в соответствующее устройство чтения-записи телефонной карты-ключа, читает и сравнивает ее содержание с закодированной картинкой, хранящейся в секретном файле. Если совпадение имеет место, программа выполняется нормально; если нет, появляется сообщение об ошибке.

Несомненно, исследователь может привнести свои изменения в алгоритм, используемый PROTECT.BAS, при условии, что точно так же будет изменен и модуль защиты основной программы. Самым простым действием было бы выполнение операции Исключающее ИЛИ, но в качестве альтернативы можно придумать секретный файл или пойти еще на какую-нибудь хитрость.

T2G, ТЕЛЕФОННАЯ КАРТА ВТОРОГО ПОКОЛЕНИЯ

Начиная с 1983 г. полупроводниковая индустрия сделала крупный рывок вперед. До сих пор технология n-МОП используется при создании чипов для современных телефонных карт (T1G). Однако она уже начинает устаревать, и изготовители должны постепенно перейти к ее замене.

Так, с 1989 г. в компании FRANCE TELECOM решили перейти на КМОП-технологию, чтобы улучшить функциональные возможности продукта, вместе с тем не увеличивая его стоимости. После проведенных экспериментов «на местности» с партией в сто тысяч карт в конце 1993 г. модернизация всего парка таксофонов уже завершена.

Эта операция была проведена без ведома большинства пользователей и, как следствие, карты T1G, еще находящиеся в обращении, в конце концов устареют и станут негодными для использования, хотя для них не существует срока годности и оплачиваются они заранее.

Замена используемой сейчас памяти ППЗУ на специфическое ЭСППЗУ, разработанное в сотрудничестве с компанией SGS-Thomson, позволяет упростить техническую базу (шесть контактов вместо восьми, отсутствие внешнего напряжения программирования Vpp) и повысить качество продукта, то есть придать ему большую гибкость. Например, допустимо запрограммировать некоторые T2G таким образом, что они смогут служить для звонка только по одному-единственному номеру.

T2G представляют собой карты с возможностью повторной записи (а в будущем, надо полагать, и повторной зарядки) и могут содержать тысячи единиц. Они уже адаптированы к посекундной оплате, которая, хочется надеяться, будет также применяться в таксофонах.

Благодаря механизму определения подлинности при помощи сертификата и вычислению подписи с помощью внутренних секретных ключей T2G являются более надежными, чем T1G, определенные возможности «клонирования» которых, как известно, представляют слабое место в безопасности системы. Нет ничего удивительного в том, что подробные технические характеристики T2G держатся в секрете.

Информация, приведенная ниже, любезно предоставлена фирмой SGS-Thomson и относится к исполнению ST 1333 (в картах T2G применяются исполнения, называемые ST 1332 или ST1303. Различия минимальны и не создают проблем в процессах чтения и даже записи). На рис. 4.7 воспроизводится картография памяти ST 1333, на базе которой велись исследования. Их результаты показаны в табл. 4.1.

Очевидно, что адресное пространство T2G больше по сравнению с T1G (счетчик адресов переполняется только на 512-м бите). Зона, представляющая наибольший интерес для чтения, совпадает с первыми 256 битами. Это значит, что можно воспользоваться форматом файлов CAR и теми же самыми устройствами чтения-записи и программным обеспечением для чтения карт (`MINILECT.BAS` и, конечно, `CARTES.EXE`).

Табл. 4.2 показывает, что на самом деле используемый протокол связи (так называемый «6 контактов») совместим с протоколом T1G. Появляется микрокоманда сравнения битов, и можно подумать, что она имеет какое-то отношение к процедуре криптографического обеспечения безопасности. К сожалению, по этой теме автору удалось получить очень мало информации.

По всей видимости, эта система близка одновременно к криптографическим алгоритмам DES и RSA посредством подписи длиной 4 бит и секретного ключа аутентификации длиной 64 бита, который

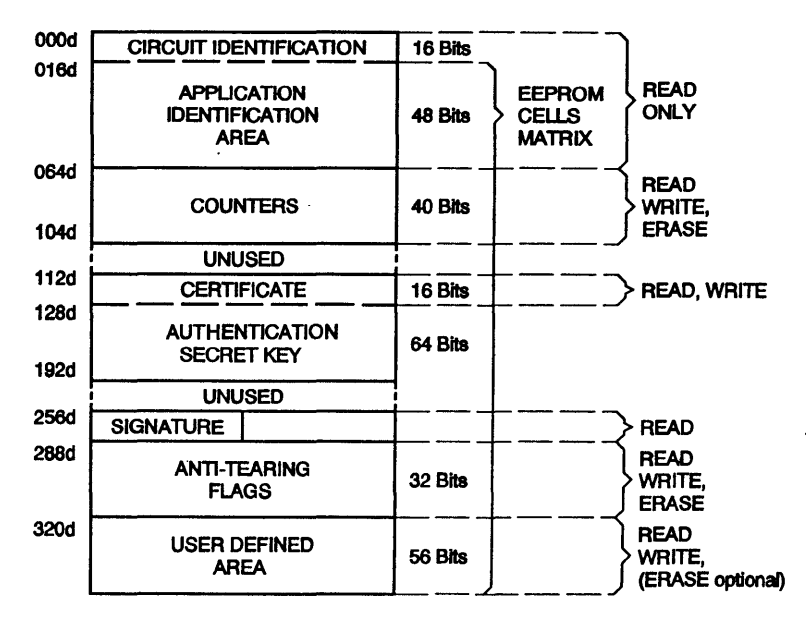


Рис. 4.7. Картография памяти ST 1333

невозможно прочитать извне, поскольку он маскируется 1. В системе также применяется «фантомное» ОЗУ, доступное для записи по адресам от 0 до 31 и для чтения по адресам от 192 до 223 или с 224 до 255. Запись в это ОЗУ влечет за собой изменение подписи (адреса с 256 до 259). С другой стороны, должны использоваться специфические программы анализа, поскольку единицы услуг рассчитываются совершенно иным методом.

За кодом Pro-Electron длиной 16 бит, который на сегодняшний день, судя по всему, является общим для всех T2G (8140h), следуют 48 бит, включающих, в частности, серийный номер карты и ее *финансовые возможности* (закодированную информацию о числе единиц, на которое была заряжена карта: 05h для 50 единиц или 0Ch для 120). Затем идут 40 бит, организованных в «счетчики» по 8 бит каждый, по «принципу счета», действующему аналогично бухгалтерским счетам (или древнему абаку). Эти 40 бит теоретически могут позволить обсчитывать до 32767 единиц!

Технология ЭСППЗУ имеет замечательную способность к повторной записи: логический 0 может быть трансформирован в логическую 1 и наоборот. Когда речь идет о картах T2G, то извне можно

Таблица 4.1. Предполагаемое содержание T2G

0	Зона записи в «фантомное» ОЗУ								31
0	1 0 0 0 Код	0 0 0 1 Proelectron	0 1 0 0	0 0 0 0	Номер серии (начало)				31
32	Номер серии (конец), всего 9 цифр					Служебный код (1)	0 (2)	0 0 0 Финансовое состояние (3)	63
64	0 0 0 0 0 0 0 0		Счетчик ×64 ед.		Счетчик ×8 ед.	1 (4)	Счетчик ×1 ед.	1 (4)	95
96	(5) 1 1 1 1 1 1 1		1 1 1 1	1 1 1 1	Сертификат (16 бит)				127
128	1 - - - Маска секретного ключа (64 бита) - - - 1								159
160	1 - - - Маска секретного ключа (читается как 64 лог. 1) - - - 1								191
192	Зона чтения «фантомного» ОЗУ								223
224	Зона чтения «фантомного» ОЗУ								255
256	Сигнатура	0 1 0 1 (6)	1 1 1 1	0 1 0 1	Флаги счетчика ×64 ед.		Флаги счетчика ×8 ед.		287
288	1 1 1 1	0 1 0 1	1 1 1 1	0 1 0 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	319
320	Зона свободной записи (0 → 1) (7)								351
352	Зона свободной записи (0 → 1)				0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	383
384	1 1 1 1	0 1 0 1	1 1 1 1	0 1 0 1 1 1 1 1	1 - - - 1				415
416	1 1 1 1	0 1 0 1	1 1 1 1	0 1 0 1 1 1 1 1	1 - - - 1				447
448	0 1 - - - 0 1								47 9
480	0 1 - - - 0 1								51 1

Примечание к табл. 4.1:

(1) Служебный код: 0000 = T2G STANDARD, 1001 = T2G EUROSTAR и т.д.

(2) № ключа

(3) Финансовые возможности: 120 единиц = 1100 (12), 50 единиц = 0101 (5), 5 или 10 единиц: 0001 (1), и т.д.

(4) Счетчики ×1 и ×8: бит справа всегда в состоянии лог. 1

(5) 0 в течение срока действия, запись с заемом стирает сертификат

(6) Плавкие перемычки

(7) Исключительно на самых современных T2G (в противном случае копируется предыдущая строчка), перечень номера(-ов)

Таблица 4.2. Предполагаемый набор мини-команд карты T2G

ISO2	ISO4	ISO3	Микрокоманда
0	0	_|‾	RESET (СБРОС)
1	0	_|‾	Не используется
0	1	_|‾	ЧТЕНИЕ (ВВЕРХ)
0	1	_|‾|_	СРАВНЕНИЕ
1	1	_|‾	ЗАПИСЬ «1» (Vcc на ISO7)

Протокол «6 контактов»: T2G, GPM256

без проблем трансформировать 0 в 1, расходуя единицы. Однако привилегия обратной трансформации (1 в 0) принадлежит только самой карте под контролем ее внутренней логики.

Безусловно, на сегодняшний день эта операция служит совсем не для перезарядки карты. Теперь надо «дебетовать» то, что в ней «накредитовано», то есть подробнее рассмотреть процессы в карте.

Каждый бит определенного счетчика имеет величину, выраженную в единицах, в восемь раз большую, чем бит счетчика, который следует за ним в порядке адресов. Хитрость состоит в том, чтобы одновременно сделать один бит счетчика равным логической 1, а восемь бит другого счетчика, следующего за ним, равными логическому 0. На практике для T2G на 120 единиц вполне достаточно трех счетчиков: один – на 1, второй – на 8 и третий – на 64. Необходимо отметить, что младший разряд двух младших счетчиков неизменно установлен в 1.

В программе T2G.BAS применяется, таким образом, соответствующий механизм для выявления состояния кредита карты на основе считанного с нее файла в формате CAR:

```
10 REM -- T2G.BAS --
20 KEY OFF:CLS
30 PRINT"Проанализировать название CAR-файла... ";
40 INPUT N$
50 IF N$="" THEN END
60 FOR F=1 TO LEN(N$)
70 IF MID$(N$,F,1)="." THEN 100
80 NEXT F
90 N$=N$+".CAR"
100 OPEN N$ FOR INPUT AS #1
110 A=0
120 FOR F=0 TO 15
130 INPUT#1,Q
140 IF Q=1 THEN A=A+2^(15-F)
150 NEXT F
```

```
160 IF A<>2^15+2^8+2^6 THEN 480
170 FOR F=16 TO 59
180 INPUT#1,Q
190 NEXT F
200 A=0
210 FOR F=60 TO 63
220 INPUT#1,Q
230 IF Q=1 THEN A=A+2^(63-F)
240 NEXT F
250 PRINT"T2G на";A*10;"единиц, ";
260 U=0
270 FOR F=64 TO 71
280 INPUT#1,Q
290 NEXT F
300 FOR F=72 TO 79
310 INPUT#1,Q
320 IF Q=1 THEN U=U+64
330 NEXT F
340 FOR F=80 TO 86
350 INPUT#1,Q
360 IF Q=1 THEN U=U+8
370 NEXT F
380 INPUT#1,Q
390 FOR F=88 TO 94
400 INPUT#1,Q
410 IF Q=1 THEN U=U+1
420 NEXT F
430 U=(10*A)-U
440 PRINT"Кредит: ";U;" UTC"
450 IF U=0 THEN PRINT:PRINT"(Кредит исчерпан)"
460 IF U<>0 THEN BEEP
470 PRINT:CLOSE#1:GOTO 30
480 BEEP:PRINT"T2G не опознана!"
490 GOTO 470
500 REM (c)1995,1997 Patrick GUEULLE
```

Разумеется, ее функциональные возможности были также включены в CARTES.EXE. Тем не менее ничто не гарантирует этим программам «вечной жизни», поскольку могут быть разработаны специальные карты T2G в связи с возможным внедрением таксофонов с посекундной оплатой или участием Франции в каком-нибудь проекте «общеевропейской телефонной карты».

Можно проследить за процессом счета на двух «отображениях содержимого памяти карты», приведенных в табл. 4.3. Одно соответствует новой карте T2G на 50 единиц, а другое – использованной карте T2G того же объема.

Таблица 4.3. Два «отображения содержимого памяти карты» T2G, прочитанные с помощью программы MINILECT.BAS

T2G 50 единиц новая (№ 200044627)

1000	0001	0100	0000	0010	0000	0000	0001	
0100	0100	0110	0010	0111	0000	0000	0101	
0000	0000	0000	0000	0000	0001	0000	0111	
0111	1111	1111	1111	0011	0001	0110	0100	Сертификат
1111	1111	1111	1111	1111	1111	1111	1111	certificat
1111	1111	1111	1111	1111	1111	1111	1111	
0000	0000	0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	0000	0000	

T2Q 50 единиц исчерпанная (№ 200141909)

1000	0001	0100	0000	0010	0000	0000	0001	
0100	0001	1001	0000	1001	0000	0000	0101	
0000	0000	0000	0111	0111	1111	0000	0111	
0111	1111	1111	1111	0001	0000	0001	1001	Сертификат
1111	1111	1111	1111	1111	1111	1111	1111	certificat
1111	1111	1111	1111	1111	1111	1111	1111	
0000	0000	0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	0000	0000	

T2G: неиспользованные единицы – «0» (карта «полная в 0»)

Для большей ясности были выделены различные счетчики, из которых два находятся в состоянии покоя. Один из них (в настоящее время все еще на нуле), вполне возможно, однажды послужит для счета по 512, поведение второго вызывает наибольший интерес.

Во время использования карты на ней находятся семь бит, установленных в 1, и один бит – в 0 (левый). Похоже, впрочем, что этот последний бит переходит в 1, когда единицы исчерпываются, по крайней мере на позднейших картах T2G. Одновременно, судя по всему, обнуляется *сертификат*, занимающий биты с 112 до 127.

Некоторые карты, очень похожие на T2G, используют похожую процедуру, позволяющую перезагрузить в них несколько единиц. При этом в них записывается новый сертификат для аутентификации операции, а один бит – в специальный счетчик, называемый *счетчиком перезагрузок*; на этом этапе вполне логично провести некоторые сравнения.

Самое удивительное, что на такой карте, считающейся очень надежной, используется стирание просроченного сертификата. Эта

операция позволяет любому лицу записать любую информацию на место прежней...

Начиная с адреса 256, то есть вне области, которую обычно считывает программа, предназначенная для T1G, расположена «подпись» на 4 бита. Она мгновенно изменяется при изменении любого бита карты, включая и те, что расположены в «фантомном» ОЗУ. Благодаря «модулю безопасности», представляющему собой чип-карту, таксофон может в любой момент определить, какую «подпись» должна выдать телефонная карта.

Естественно, любое несоответствие между тем, что ожидалось, и конечным результатом интерпретируется как попытка мошенничества, даже если реальной причиной послужило состояние электрических контактов. Необходимо, тем не менее, отметить, что фальшивая подпись имеет один шанс из шестнадцати сойти за правильную. Это объясняется тем, что четыре бита могут определить только шестнадцать различных комбинаций.

Интересно проследить, как изменяется значение этой криптографической подписи, не пытаясь проникнуть во все ее секреты. Например, можно заметить, что она меняется не сразу, а при переходе через ноль счетчика адресов.

Для того чтобы провести некоторые наблюдения, иногда достаточно небольшой программы SIGNT2G.BAS.

```
10 REM -- SIGNT2G.BAS --
20 KEY OFF:CLS:DEF SEG=0
30 S1=PEEK(&H408)+256*PEEK(&H409) 'для LPT1:
40 S2=PEEK(&H40A)+256*PEEK(&H40B) 'для LPT2:
50 OUT S2,0:E2=S2+1
60 IF (INP(E2) AND 64) <> 0 THEN S=S1:GOTO 100
70 OUT S2,128
80 IF (INP(E2) AND 64) <> 64 THEN S=S1:GOTO 100
90 S=S2
100 E=S+1
110 OUT S,0:KEY OFF:CLS
120 PRINT"Вставить карту T2G, затем нажать ENTER ";
130 INPUT Z$:PRINT:PRINT
140 OUT S,250:OUT S,248
150 FOR F=0 TO 255
160 OUT S,249:OUT S,251
170 NEXT F
180 FOR F=1 TO 4
190 GOSUB 260
200 NEXT F
210 PRINT
220 FOR F=260 TO 511
```

```
230 OUT S,249:OUT S,251
240 NEXT F
250 GOTO 150
260 OUT S,249
270 D=INP(E):GOSUB 320
280 Z$=INKEY$
290 IF Z$=CHR$(27) THEN OUT S,0:END
300 OUT S,251
310 RETURN
320 K= (D AND 128)
330 IF K<>128 THEN PRINT"1";
340 IF K=128 THEN PRINT"0";
350 RETURN
360 REM (c)1995 Patrick GUEULLE
```

Создается впечатление, что даже если содержание карты остается прежним, подпись меняется при каждом «полном цикле» счетчика, и полученные результаты подтверждают это. Значения подписи повторяются, так как для нее возможны лишь шестнадцать различных комбинаций.

Эта особенность позволяет использовать израсходованную карту T2G в качестве генератора простейших ключей для простых криптографических приложений. Основной ключ можно было бы время от времени менять путем записи одного бита в зону счетчиков или сертификата.

В той же самой области, что и ключ, есть несколько битов (возможно, четыре), повторяющих состояние внутренних «плавких перемычек», которое зависит от различных этапов «жизненного цикла» карты, начиная с ее персонализации и вплоть до конечной фазы использования.

Немного дальше находятся *флажки против вырывания*. Эти биты служат для исключения потери единиц услуг владельцем карты в случае ее вытаскивания до завершения процесса «заема» – списывания одной единицы услуг.

Каждому биту счетчиков по 64 и по 8 соответствует один бит, или флаг, занимающий соответствующее положение в одном из двух регистров против вырывания. Обычно флаг установлен в состояние 0, в момент «заема» трансформируется в 1 и опять возвращается в 0 одновременно с изменением состояния счетчика младших разрядов. Если карта «вырвана» между этими двумя операциями, то флаг, разумеется, остается в 1 и разрешает во время следующего вставления карты обнуление счетчика без фиксации «заема». С точки зрения бухгалтерии, инцидент исчерпан.

По желанию можно запустить данный механизм, который при обычных условиях работает только в исключительных случаях. Для этого намеренно укорачивают импульс, служащий для выполнения операции «записи с заемом» (см., например, программу MANIPT2G.BAS, которая описана в книге «Чип-карты. Устройство и применение в практических конструкциях» и приведена на сайте www.dmk.ru).

Нет необходимости уточнять, что флаги против вырывания могут быть трансформированы в 1 только с помощью внутренней логики карты и ни в коем случае не по команде извне.

Продолжая исследование памяти T2G последних выпусков, можно найти зону с 320 до 367, свободную для записи (преобразование 0 в 1). В некоторых картах, родственных T2G, в этой зоне разрешается производить и стирание (преобразование 1 в 0). Похоже, что такая область памяти предназначена для занесения одного или двух номеров телефонов, по которым с помощью данной карты можно звонить автоматически.

В принципе адресуемое пространство T2G здесь заканчивается, но адресный счетчик продолжает считать до 512. Чтение зоны, расположенной между этими двумя адресами, по всей вероятности, не имеет большого значения, хотя и там наблюдаются группы нулей и единиц.

ЕВРОПЕЙСКИЕ КАРТЫ

В Германии чип-карты появились намного позже, чем во Франции, поэтому при разработке Telefonkarte использовались более передовые технологии.

Чип компании Siemens (второй поставщик – Philips), снабженный памятью ЭСППЗУ, ныне имеет многочисленные аналоги. Он предвосхитил появление разработанной во Франции карты T2G (которая, однако, уже снабжена криптографической системой безопасности). Его последняя версия Eurochip на сегодняшний день обладает всеми функциями обеспечения безопасности, которые ничем не уступают подобным функциям T2G. Чип последовательно вводится в обращение во всех основных европейских странах: Германии, Голландии, Швейцарии, Великобритании и т.д.

Очень хочется думать, что Eurochip является ответом на международное распространение карты T2G, производимой исключительно компанией SGS-Thomson. Более того, здесь видится своего рода реванш за достижения французов в области чип-карт.

Но, в конце концов, ни одна из французских карт типа T1G не была снабжена совместимым чипом Siemens, хотя продукция SGS-Thomson

очень широко представлена на рынке Telefonkarte. Возможно, это и объясняет нижеследующие положения.

Сходство первой Telefonkarte с T2G очевидно на уровне картографии, особенно в том, что касается первых 104 бит, поскольку у предшественника Eurochip их ровно столько же (табл. 4.4).

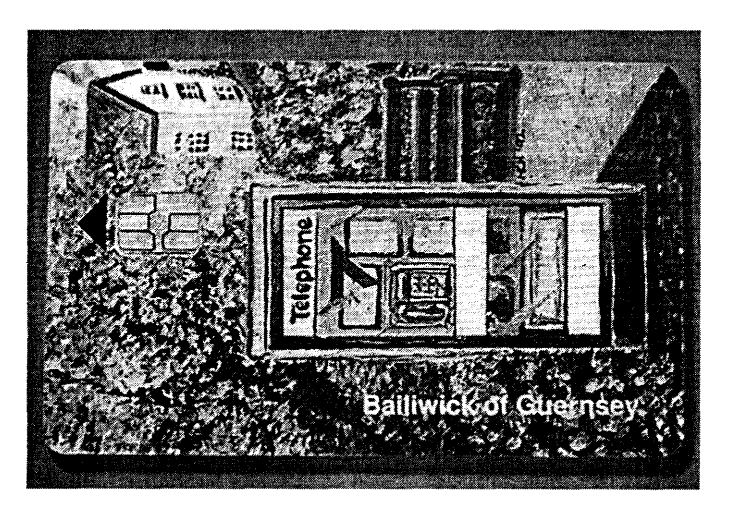


Рис. 4.8. Карта, совместимая с Telefonkarte

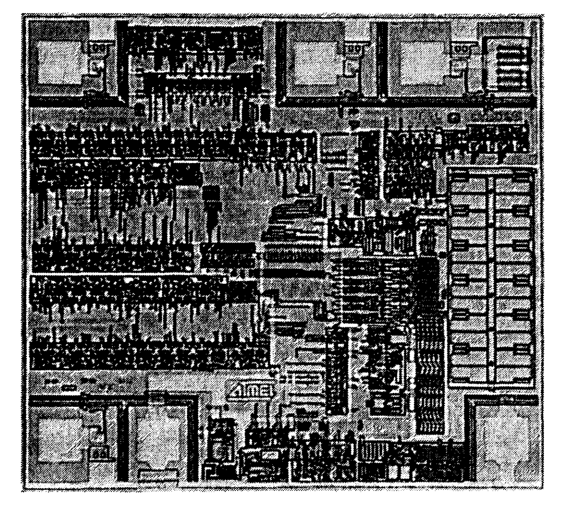


Рис. 4.9. Чип Amtel, совместимый с Telefonkarte

Таблица 4.4. Картография памяти Telefonkarte

Функция		Адрес	Биты
Данные по чипу		0–23	24
Данные по изготовителю		24–63	40
Полная персонализация		64	1
Данные по счету	Биты тестирования (ТВ)	65–66	2
	Цифра 5 (D5)	67–71	5
	Цифра 4 (D4)	72–79	8
	Цифра 3 (D3)	80–87	8
	Цифра 2 (D2)	88–95	8
	Цифра 1 (D1)	96–103	8
			104

Самый старший счетчик (D5) содержит только 5 бит, что теоретически сокращает возможности счета до 25160 единиц (хотя изготовители анонсировали 20480). Таким образом, из зоны счета исключаются два бита тестирования (b65 и b66) и один, служащий ключом при завершении операции персонализации (b64): в каком-то смысле он является программной «плавкой перемычкой».

Однако протокол связи значительно отличается от протокола T2G и даже T1G. Кроме того, стоит признать, что он больше соответствует стандартам ISO, которые только разрабатывались во время внедрения первых французских телефонных карт и с которыми T2G должны быть совместимы в определенной степени совместимы.

Из табл. 4.4 видно, что речь идет о протоколе «5 контактов» и что карта имеет тип «полная в 1»: для того, чтобы потребить единицы услуг, 1 трансформируются в 0.

Внимание! Этот протокол несовместим с условиями сброса (*reset*) карты T1G. Следовательно, если используются одни и те же программы (MINILECT.BAS или CARTES.EXE), результаты чтения будут сдвинуты на один бит. Достаточно знать об этой особенности и при необходимости учитывать ее.

Для создания серьезных приложений предпочтительно обратиться к более специфическим программам, которые можно найти в книге «Чип-карты. Устройство и применение в практических конструкциях».

Принцип работы, аналогичный тому, который применен в бухгалтерских счетах, использует предварительную установку счетчиков при выдаче карты, поэтому все их биты находятся в состоянии логического 0,

когда кредит карты исчерпан. При отсутствии всякой системы криптографической безопасности это еще одно средство против незаконной повторной зарядки карты (табл. 4.5).

Таблица 4.5. Система команд Telefonkarte

ISO2	ISO3	Микрокоманда
1	_|‾	RESET (СБРОС)
0	_|‾	ЧТЕНИЕ (ВВЕРХ)
0	_|‾|_	СРАВНЕНИЕ
_|‾|_ Затем: 0	0 _|‾	ЗАПИСЬ «0» (ISO7 на общ. проводе)

Протокол «5 контактов»: Eurochip

Два «отображения содержимого памяти карт» (табл. 4.6) состоят из 256 бит, считать которые достаточно легко.

Полезное содержание карты здесь фигурирует дважды, поскольку счетчик адресов переполняется после 128 тактовых импульсов. Эта особенность позволяет распознавать старые карты, так как на более поздние она не распространяется.

Таблица 4.6. Два «отображения содержимого памяти» карт Telefonkarte

Telefonkarte 1200 новая

1110	0000	0101	1111	1111	1110	1001	0101
0101	0000	0101	0100	0101	0001	0000	0000
0000	0000	0000	0110	0000	0110	0111	1110
0000	0001	1111	1111	1111	1111	1111	1111
1110	0000	0101	1111	1111	1110	1001	1101
0101	0000	0101	0100	0101	0001	0000	0000
0000	0000	0000	0110	0000	0110	0111	1110
0000	0001	1111	1111	1111	1111	1111	1111

Telefonkarte 1200 использованная

1110	0000	0101	1111	1111	1111	1001	0101
0100	0001	1010	1100	0100	1000	1000	0000
0000	0000	0000	0000	0000	0000	0000	0000
0000	0001	1111	1111	1111	1111	1111	1111
1110	0000	0101	1111	1111	1111	1001	0101
0100	0001	1010	1100	0100	1000	1000	0000
0000	0000	0000	0000	0000	0000	0000	0000
0000	0001	1111	1111	1111	1111	1111	1111

Telefonkarte: неизрасходованные единицы – в лог. 1
(карта «полная в 1»)

Стоит отметить использование четырех счетчиков из пяти имеющихся в наличии, поскольку считаются *денежные единицы* (пфенниги в Германии, пенсы в Великобритании или на острове Гернси и т.д.), а не телефонные звонки, как раньше. На самом деле Telefonkarte на 40 единиц содержит 1200 пфеннигов, то есть 12 марок. В этих условиях переход на посекундную оплату будет не более чем простой формальностью.

Небольшая программа EURO.BAS способна определить размер кредита европейской карты на основании ее образа, считанного при помощи программы MINILECT.BAS в виде CAR-файла:

```
10 REM -- EURO.BAS --
20 KEY OFF:CLS
30 PRINT"Имя CAR-файла ... для анализа ";
40 INPUT N$
50 IF N$="" THEN END
60 FOR F=1 TO LEN(N$)
70 IF MID$(N$,F,1)="." THEN 100
80 NEXT F
90 N$=N$+".CAR"
100 OPEN N$ FOR INPUT AS #1
110 FOR F=0 TO 66
120 INPUT#1,Q
130 NEXT F
140 U=-1
150 FOR F=67 TO 71
160 INPUT#1,Q
170 IF Q=1 THEN U=U+4096
180 NEXT F
190 FOR F=72 TO 79
200 INPUT#1,Q
210 IF Q=1 THEN U=U+512
220 NEXT F
230 FOR F=80 TO 87
240 INPUT#1,Q
250 IF Q=1 THEN U=U+64
260 NEXT F
270 FOR F=88 TO 95
280 INPUT#1,Q
290 IF Q=1 THEN U=U+8
300 NEXT F
310 FOR F=96 TO 103
320 INPUT#1,Q
330 IF Q=1 THEN U=U+1
340 NEXT F:PRINT
350 PRINT"КРЕДИТ: ";U;" денежных единиц"
360 IF U=0 THEN PRINT:PRINT"(Кредит исчерпан)"
```

```
370 IF U<>0 THEN BEEP
380 PRINT:CLOSE#1:GOTO 30
390 REM (c)1995 Patrick GUEULLE
```

CARTES.EXE делает то же самое, определяя по мере возможности «национальность» карты, ее изготовителя, валюту, в которой она функционирует.

При высоком уровне совместимости на уровне счетчиков единиц (все на 8 бит) новая карта Eurochip имеет более сложную структуру памяти, которая наглядно представлена в табл. 4.7. По данным Siemens, полезный объем памяти состоит из 237 бит, распределенных между зонами ПЗУ (из которых 16 бит запрограммированы маской), ППЗУ и ЭСППЗУ.

Исследования, проведенные автором (см. табл. 4.8), выявляют зону, свободную для записи, которая располагается между 320 и 383 битами, а также адресный счетчик, переполняющийся на бите 512. Становится ясно, что неиспользуемые зоны должны размещаться между рабочими.

Перед нами открыта широкая область для захватывающих экспериментов, так как «маленькие секреты» Eurochip еще держатся в тайне. Создается впечатление, что каждая страна, в которой используются эти телефонные карты, обладает персонализированной версией микросхемы, имеющей свои особенности.

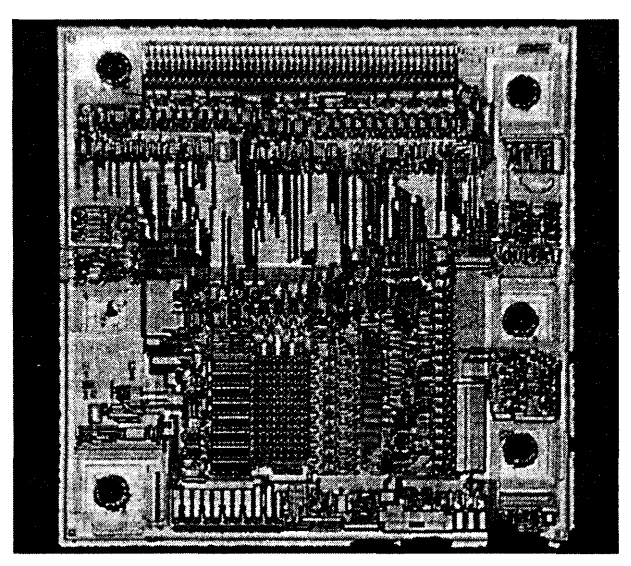


Рис. 4.10. Чип марки Siemens для карты Eurochip

Программное обеспечение чип-карт, описываемое в данной книге, представлено на сайте www.dmk.ru. Эти разработки являются результатом многолетних кропотливых исследований.

Таблица 4.7. Картография Eurochip по данным Siemens

Данные идентификации	64 бита, из которых 16 – масочного типа	ПЗУ
Счетчики	40 бит	ЭСППЗУ
Ключ 1	48 бит	ПЗУ
Флаг ключа 2	1 бит	ПЗУ
Ключ 2/зона владельца	48 бит	ПЗУ/ППЗУ
Зона владельца	16 бит	ППЗУ
Зона владельца	16 бит	ППЗУ
Флажки против вырывания	4 бита	ЭСППЗУ
	237 бит (221+16 масочного типа)	

Таблица 4.8. Суммарные результаты исследований

0	Номер кремния	Код страны		Код изготовителя	Ключ	31
32	Номер серии и другие данные (издателя или дистрибьютора)					63
64	Счетчик ×4096	Счетчик ×512	Счетчик ×64	Счетчик ×8		95
96	Счетчик ×1					127
128						159
160						191
192						223
224						255
256						287
288						319
320	Зона, свободная для записи (1 → 0)	Зона, свободная для записи (1 → 0)				351
352	Зона, свободная для записи (1 → 0)	Зона, свободная для записи (1 → 0)				383
384						415
416						447
448						479
480						511

ГЛАВА СТРАНИЦА

5 ПРОГРАММЫ И ФАЙЛЫ

Предлагаемая вашему вниманию подборка, единственная в своем роде, включает в себя различные программы, используемые в данной книге, а также некоторые другие интересные и полезные файлы.

Настоящее издание представляет собой продолжение книги «Чип-карты. Устройство и применение в практических конструкциях», где используется аналогичный подход к представлению программного обеспечения. Автор не боится утверждать, что собранная им информация и разработанные инструменты дадут возможность обладателю обоих изданий довести исследования до конца. Имеются в виду реальные проекты, на которые может рассчитывать любитель, изучающий чип-карты.

ОСОБЕННОСТИ ПРОГРАММ

Многие программы, перечисленные ниже, могут функционировать в замедленном темпе и на более старой технике (8088 на частоте 4,77 МГц), если они были скопированы на соответствующую дискету.

T2G	BAS	1 039	30.10.97	12:30	T2G.BAS
CARS232	BAS	873	05.03.95	18:56	CARS232.BAS
CB2PIN	BAS	706	05.03.95	18:51	CB2PIN.BAS
EURO	BAS	831	06.03.95	19:07	EURO.BAS
MINILECT	BAS	1 262	08.04.95	9:22	MINILECT.BAS
MODULO	BAS	284	26.03.95	11:16	MODULO.BAS
PIN2CB	BAS	695	05.03.95	18:50	PIN2CB.BAS
PROTECT	BAS	1 054	27.03.95	9:41	PROTECT.BAS
PROTEGE	BAS	1 222	27.03.95	9:37	PROTEGE.BAS
RSA	BAS	422	26.03.95	17:24	RSA.BAS
SIMU	BAS	1 044	05.03.95	18:57	SIMU.BAS
T1G	BAS	3 328	05.03.95	18:43	T1G.BAS
TEXAS	BAS	905	08.04.95	9:35	TEXAS.BAS
XOR	BAS	226	26.03.95	11:08	XOR.BAS
ADL	BAS	21 888	07.05.97	9:45	ADL.BAS
ADT	BAS	2 710	07.05.97	9:44	ADT.BAS
DECADL	BAS	3 698	24.04.97	9:27	DECADL.BAS
DECADT	BAS	1 116	25.04.97	10:41	DECADT.BAS
DIALINV	BAS	1 580	06.01.98	7:49	DIALINV.BAS
ESPDIR	BAS	587	22.04.97	9:24	ESPDIR.BAS
ESPINV	BAS	839	22.04.97	9:53	ESPINV.BAS
LECTDIR	BAS	2 285	07.05.97	9:50	LECTDIR.BAS
LECTINV	BAS	2 573	07.05.97	9:49	LECTINV.BAS
SIGNT2G	BAS	858	24.05.95	18:22	SIGNT2G.BAS
DIALDIR	BAS	1 292	06.01.98	7:52	SIGNT2G.BAS
OPDIR	BAS	1 514	05.01.98	17:12	OPDIR.BAS
OPINV	BAS	1 827	05.01.98	17:11	OPINV.BAS
INSTALL	EXE	30 733	05.04.95	16:28	INSTALL.EXE
ADL	EXE	29 389	08.01.98	9:36	ADL.EXE

ADT	EXE	31 917	08.01.98	9:36	ADT.EXE
DECADL	EXE	31 101	08.01.98	9:36	DECADL.EXE
DECADT	EXE	31 725	08.01.98	9:36	DECADT.EXE
DIALINV	EXE	29 501	08.01.98	9:36	DIALINV.EXE
ESPDIR	EXE	28 221	08.01.98	9:37	ESPDIR.EXE
ESPINV	EXE	28 365	08.01.98	9:37	ESPINV.EXE
LECTDIR	EXE	29 645	08.01.98	9:37	LECTDIR.EXE
LECTINV	EXE	29 805	08.01.98	9:37	LECTINV.EXE
CARTES	EXE	35 325	08.01.98	9:38	CARTES.EXE
CARS232	EXE	28 749	08.01.98	9:38	CARS232.EXE
SIMU	EXE	28 829	08.01.98	9:38	SIMU.EXE
DIALDIR	EXE	29 325	08.01.98	9:36	DIALDIR.EXE
OPDIR	EXE	29 453	08.01.98	9:38	OPDIR.EXE
OPINV	EXE	29 629	08.01.98	9:39	OPINV.EXE
INVERSE	EXE	28 877	08.01.98	9:39	INVERSE.EXE
DIRECT	EXE	28 717	08.01.98	9:39	DIRECT.EXE
PICPUCE	ASM	5 565	24.03.95	8:11	PICPUCE.ASM
COUP84	ASM	2 722	18.04.97	11:43	COUP84.ASM
COUP84	HEX	6 179	18.04.97	11:50	COUP84.HEX
PICPUCE	HEX	6 179	02.01.98	18:25	PICPUCE.HEX
PICPUCE	OBJ	960	24.03.95	11:13	PICPUCE.OBJ
COUP884	OBJ	420	18.04.97	11:44	COUP84.OBJ
COUP	PCB	3 242	17.04.97	10:16	COUP.PCB
ESPION	PCB	1 822	22.04.97	11:05	ESPION.PCB
BITT	PCB	1 225	30.10.97	13:54	BITT.PCB
MINILECT	PCB	1 078	30.10.97	13:45	MINILECT.PCB
AFNOR	PCB	884	30.11.97	13:56	AFNOR.PCB
PICISO	PCB	1 039	30.11.97	13:57	PICISO.PCB
PICAFNOR	PCB	1 012	30.11.97	13:57	PICAFNOR.PCB
PICPUCE	PCB	636	30.11.97	13:58	PICPUCE.PCB
ALM1	CAR	592	24.11.94	17:43	ALM1.CAR
ALM2	CAR	592	24.11.94	17:44	ALM2.CAR
ALM3	CAR	592	28.03.95	13:15	ALM3.CAR
ALM4	CAR	592	28.03.95	13:15	ALM4.CAR
GUERN1	CAR	592	16.08.94	12:03	GUERN1.CAR
T2G5	CAR	528	10.03.95	18:29	T2G5.CAR
T2G6	CAR	528	10.03.95	18:30	T2G6.CAR
T2GA	CAR	592	10.03.95	18:05	T2GA.CAR
T2G4	CAR	592	10.03.95	18:05	T2G4.CAR
GUERN2	CAR	592	16.08.94	12:03	GUERN2.CAR
HOLA	CAR	592	10.03.95	18:07	HOLA.CAR
IRLANDE	CAR	592	10.03.95	18:06	IRLANDE.CAR
MALTE	CAR	592	30.08.94	11:09	MALTE.CAR
MOBIL	CAR	592	28.03.95	13:13	MOBIL.CAR
PASVIDE	CAR	592	23.10.94	15:59	PASVIDE.CAR
CARDSPEC	PDF	738 275	03.09.97	12:32	CARDSPEC.PDF
CARDSPEC	(C)	1 705	08.01.98	9:57	CARDSPEC.C
	76	файл(ов)	1 357 959 байт		
	0	репертуар(ов)	75 776 байт свободных		

Все программы для ПК представлены на языке GWBASIC (файлы с расширением .BAS), а большинство из них продублировано исполняемыми файлами DOS (расширение .EXE).

Чаще всего предпочтение отдается последнему варианту, но в случае возникновения особых проблем можно воспользоваться текстом на BASIC.

Программы, предназначенные для микроконтроллера PIC16C84, приводятся в виде исходных текстов (расширение .ASM) и файлов двух других типов для его «прошивки».

Файлы .OBJ в наибольшей степени подходят для программатора PICSTART 16B производства компании Microchip, а файлы .HEX предназначены главным образом для очень простого программатора, собранного в соответствии со схемой, которая приведена в книге «Как превратить персональный компьютер в универсальный программатор».

Все печатные платы, топология которых представлена в данной книге, представлены в виде файлов в формате PCB, совместимых с пакетом программ BOARDMAKER. На компакт-диске, прилагаемом к книге “Logiciels PC pour l’electronique”[1], содержится сокращенная версия BOARDMAKER, позволяющая загружать, визуализировать, изменять по желанию и перепечатывать эти файлы. Однако их запись на диск не предусмотрена.

Также представлено несколько «отображений содержимого памяти карт» (CAR-файлы) с тем, чтобы дать исследователю возможность опробовать предлагаемые программы даже при отсутствии устройства чтения-записи или телекарты.

ALM1.CAR и ALM4.CAR получены от четырех различных (пустых и начатых) немецких телефонных карт. GUERN1.CAR и GUERN2.CAR были считаны с двух карт острова Гернси – пустой и начатой, – совместимых с немецкой технологией (Telefonkarte). MALTE.CAR и IRLANDE.CAR, соответственно, относятся к двум картам Мальты и Ирландии (та же технология, что у французской T1G, но с другой кодировкой).

HOLA.CAR (ЭЙ_ТАМ.CAR) и PASVIDE.CAR (НЕПУСТАЯ.CAR) считаны с двух французских телефонных карт, которые имеют различные «коды семейства»: 03h для PASVIDE.CAR и 04h для HOLA.CAR.

До сих пор еще встречаются коды до 07h, и только окончательный переход к технологии T2G сможет остановить это бесконечное развитие, обусловленное огромными объемами выпуска карт.

---

[1] Русский перевод («Программы для радиолюбителей») готовится к выпуску в издательстве «ДМК».

T2GA.CAR и T2G5.CAR происходят от двух телефонных карт второго поколения, над которыми проводились эксперименты в 1993 году. Одна была новая, а вторая содержала еще пять из пятидесяти единиц, заряженных на заводе.

MOBIL.CAR является образцом карты, которая используется для оплаты мытья машин на станциях автосервиса MOBIL или BP. Есть варианты на 12 и 24 единицы (раньше они выпускались на 10 и 20 единиц соответственно). Здесь применяется технология (Solaic) карт первого поколения T1G.

```
ALM1.CAR
0000 0010 0011 1111 1111 1110 1010 1010
1000 1011 0010 0101 0011 0011 0001 1000
0000 0000 0000 0000 0000 0000 0000 0100
0000 1111 1111 1111 1111 1111 1111 1111
0000 0010 0011 1111 1111 1110 1010 1010
1000 1011 0010 0101 0011 0011 0001 1000
0000 0000 0000 0000 0000 0000 0000 0100
0000 1111 1111 1111 1111 1111 1111 1111
ALM2.CAR
1101 0000 1011 1111 1111 1110 0010 1010
1001 0011 1010 0010 0101 1001 1010 0000
0000 0000 0000 0000 0011 1100 0000 0100
1111 1111 1111 1111 1111 1111 1111 1111
1101 0000 1011 1111 1111 1110 0010 1010
1001 0011 1010 0010 0101 1001 1010 0000
0000 0000 0000 0000 0011 1100 0000 0100
1111 1111 1111 1111 1111 1111 1111 1111
ALM3.CAR
1110 0000 0101 1111 1111 1110 1001 0101
0101 0000 0101 0100 0101 0001 0000 0000
0000 0000 0000 0110 0000 0110 0111 1110
0000 0001 1111 1111 1111 1111 1111 1111
1110 0000 0101 1111 1111 1110 1001 0101
0101 0000 0101 0100 0101 0001 0000 0000
0000 0000 0000 0110 0000 0110 0111 1110
0000 0001 1111 1111 1111 1111 1111 1111
ALM4.CAR
1110 0000 0101 1111 1111 1111 1001 0101
0101 0001 1000 1001 1101 1001 1000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0001 1111 1111 1111 1111 1111 1111
1110 0000 0101 1111 1111 1111 1001 0101
0101 0001 1000 1001 1101 1001 1000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0001 1111 1111 1111 1111 1111 1111
```

```
GUERN1.CAR
0010 1000 0111 0111 1111 1110 0000 1100
1110 0000 0000 0000 0000 0011 1011 1100
0000 0000 0000 0000 0000 0000 0001 1110
0001 1111 1111 1111 1111 1111 1111 1111
0010 1000 0111 0111 1111 1110 0000 1100
1110 0000 0000 0000 0000 0011 1011 1100
0000 0000 0000 0000 0000 0000 0001 1110
0001 1111 1111 1111 1111 1111 1111 1111
GUERN2.CAR
0010 1000 0111 0111 1111 1110 0000 1100
1110 0010 0000 0011 0101 1000 1100 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0001 1111 1111 1111 1111 1111 1111
0010 1000 0111 0111 1111 1110 0000 1100
1110 0010 0000 0011 0101 1000 1100 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0001 1111 1111 1111 1111 1111 1111
HOLA.CAR
1101 0011 0000 0100 0000 0000 0001 0110
1100 1111 0010 0000 0110 1000 0001 0000
1100 0111 1000 1011 0001 0000 0000 0110
1111 1111 1111 1111 1111 1111 1111 1111
1111 1111 1111 1111 1111 1111 1111 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 1111 1111
IRLANDE.CAR
1011 1010 1000 0011 0001 0001 0000 0010
0100 0000 0000 0001 1100 1101 0100 1010
0001 1001 1010 0111 0001 0001 0011 1100
1110 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
MALTE.CAR
1011 1101 1000 0011 0001 0000 0110 0010
0000 0000 0001 0101 1010 0100 0101 1010
1000 1000 1001 0100 0001 0001 0101 0100
1111 1111 1111 1111 1111 1111 1111 1111
1111 1111 1111 1111 1111 1111 1111 1100
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
```

```
MOBIL.CAR
1000 1000 1000 0000 0010 0000 0000 0010
0011 1100 0111 0101 1000 0010 0010 0100
1010 0001 0000 0000 0000 0000 0000 0001
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
PASVIDE.CAR
1100 1011 0000 0011 1001 0000 0100 0001
1011 1011 0101 0110 0110 0111 0000 0010
1100 0011 0100 1111 0001 0000 0000 0110
1111 1111 1111 1111 1111 1111 1111 1111
1111 1111 1111 1111 1111 1111 1110 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
T2G5.CAR
1000 0001 0100 0000 0010 0000 0000 0000
1000 0010 0100 0101 0110 0000 0000 0101
0000 0000 0000 0000 0011 1111 0011 1111
0111 1111 1111 1111 0101 0011 1000 0010
1111 1111 1111 1111 1111 1111 1111 1111
1111 1111 1111 1111 1111 1111 1111 1111
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
T2GA.CAR
1000 0001 0100 0000 0010 0000 0000 0001
0100 0010 0101 0010 1000 0000 0000 0101
0000 0000 0000 0000 0000 0001 0000 0001
0111 1111 1111 1111 0111 1000 1011 1010
1111 1111 1111 1111 1111 1111 1111 1111
1111 1111 1111 1111 1111 1111 1111 1111
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
```

Кроме этого, к книге прилагается документ, представляющий исключительный интерес для наиболее просвещенных читателей.

CARDSPEC.PDF полностью воспроизводит спецификацию EMV – карт для ведения денежных расчетов, которые войдут в употребление в будущем. Она читается при помощи бесплатной программы Acrobat Reader, которая находится на компакт-диске, прилагаемом к книге «Современные микроконтроллеры».

Данный документ, размещенный на сервере компании VISA, представленный на английском языке (187 страниц), широко проиллюстрирован и содержит основные положения стандарта ISO 7816. Воспроизведение и копирование этой информации разрешено при соблюдении условий и положений, изложенных в текстовом файле CARDSPEC.(C).

ИНСТАЛЛЯЦИЯ ПРОГРАММ

В основном все программы на языке BASIC могут быть использованы сразу, однако для выполнения CARTES.EXE необходимо провести очень простую инсталляцию.

Данная операция требует предварительной реализации устройства чтения-записи для телефонных карт (например, описанного в главе 4). Это необходимо для «связывания» дискеты с любой телефонной картой-ключом. Подобную инсталляцию автор рекомендует применять только при изготовлении копии исключительно для личного пользования.

Любое воспроизведение оригинальной дискеты, не сопровождаемое телефонной картой-ключом, использованной при инсталляции, будет считаться «пиратской» копией. Ее использование преследуется по закону.

Инсталляция должна проводиться следующим образом:

1. Соответствующее устройство чтения-записи для телефонных карт подключить к параллельному порту ПК (LPT1 или LPT2) и запустить сервисную программу INSTALL.EXE, набрав INSTALL с оригинальной дискеты, не защищенной от записи.
2. По требованию программы вставить любую телефонную карту в устройство чтения-записи, затем нажать ENTER. Через несколько секунд на экране появится значок авторского права. Это значит, что теперь копия файла записана на дискете, которую можно повторно защитить от записи перед тем, как сделать копию для личного пользования.
3. Запустить CARTES.EXE, набрав CARTES, и по запросу программы вставить телефонную карту-ключ.
4. Если выводится сообщение «Карта-ключ не опознана» или другая информация об ошибке, надо тщательно проверить устройство чтения-записи, запустив из интерпретатора GWBASIC программу MINILECT.BAS.
5. Если сбой касается только CARTES.EXE, необходимо повторно начать инсталляцию с другой телефонной картой.

6. После завершения инсталляции желательно полностью скопировать содержимое дискеты в директорию на жестком диске или, в крайнем случае, на рабочую дискету.

Внимание! При копировании одной программы CARTES.EXE она не будет работать.

СПОСОБ ИСПОЛЬЗОВАНИЯ ПРОГРАММЫ CARTES.EXE

Указанная программа конкретизирует всю информацию, собранную на момент написания данной книги, о французских и других телефонных картах. Эти сведения неокончательны.

CARTES.EXE должна сочетаться с устройством чтения-записи синхронных карт (см. главу 4), которое инсталлировано вышеописанным способом. Разумеется, телефонная карта-ключ должна представляться при каждом запуске программы (желательно хранить ее вместе с копией).

Программа предназначена в основном для коллекционеров и любителей и не предусматривает записи на телефонных картах. Операция такого рода требует использования более сложного устройства чтения-записи и специфического программного обеспечения, которые описаны в книге «Чип-карты. Устройство и применение в практических конструкциях».

В предлагаемом исполнении CARTES.EXE воспринимает следующие семейства синхронных карт:

- карты ППЗУ на 256 бит GPM256: французские T1G, телефонные карты многих стран, карты разного назначения (для оплаты мытья машин, платы за паркинг, посещения кинотеатров и т.д.);
- французские T2G на 50 и 120 единиц;
- европейские карты и Eurochip (Германия, Голландия, Швейцария, остров Гернси, Великобритания, некоторые частные карты и т.д.).

Автоматическая идентификация страны и считывание оставшихся в наличии единиц предусматривается в тех случаях, когда информация это позволяет. В противном случае (карты Испании и Хорватии, например, сложнее заставить «говорить»...) указывается, чем эти данные могли бы быть.

Так или иначе, опция отображения в двоичной и шестнадцатеричной системе позволяет исследовать «отображение содержимого памяти карты» в необработанном, недекодированном виде. Таким образом, при желании каждый может попытаться проникнуть в тайны чип-карт.

Патрик Гёлль

ПК и чип-карты

Главный редактор *Захаров И. М.*
editor-in-chief@dmkpress.ru
Перевод с фр. *Сомова Н. О.*
Научный редактор *Бряндинский А. Э.*
Литературный редактор *Готлиб О. В.*
Технический редактор *Прока С. В.*
Верстка *Тарасов С. А.*
Графика *Бахарев А. А.*
Дизайн обложки *Антонов А. И.*

Подписано в печать 12.04.2003. Формат 60×88 1/16
Гарнитура «Петербург». Печать офсетная.
Усл. печ. л. 8,82. Тираж 500 экз.
Зак. № 570

Издательство «ДМК Пресс», 105023, Москва, пл. Журавлева, д. 2/8.
Web-сайт издательства: www.dmkpress.ru.
Internet-магазин: www.abook.ru.

Отпечатано в типографии № 9,
Волочаевская, 40.